



FlyCapture 2.5

Managed API Programming Reference

Revised April 13, 2013

Point Grey Research Inc.

12051 Riverside Way • Richmond, BC • Canada • V6W 1K7 • T (604) 242-9937 •
www.ptgrey.com

Software Warranty

Point Grey Research warrants to the Original Purchaser, for a period of one (1) year from date of purchase that:

1. The diskette on which the Software is furnished and the accompanying documentation are not defective;
2. The Software is properly recorded upon the diskettes enclosed;
3. The documentation is substantially complete and contains all the information Point Grey Research deems necessary to use the Software;
4. The Software functions substantially as described in the documentation.

Point Grey Research, Inc.'s entire liability and the Original Purchaser's exclusive remedy shall be the replacement of any diskette or documentation not meeting these warranties. On such an occasion, a copy of the paid receipt accompanied with the faulty diskette or documentation must be returned to Point Grey Research, Inc. or an authorized dealer.

Point Grey Research, Inc. expressly disclaims and excludes all other warranties, express, implied and statutory, including, but without limitation, warranty of merchantability and fitness for a particular application or purpose. In no event shall Point Grey Research, Inc. be liable to the Original Purchaser or any third party for direct, indirect, incidental, consequential, special or accidental damages, including without limitation damages for business interruption, loss of profits, revenue, data or bodily injury or death.

Software License Agreement

READ CAREFULLY: This is a legal agreement between you (an individual or a single entity) ("you") and Point Grey Research, Inc. ("PGR"). Before installing and using the FlyCapture® Software Development Kit and any updates to it that we may at our discretion provide to you (collectively, the "SDK"), you should read this agreement. If you do not agree with all of the terms of this agreement, do not install or use the SDK. PGR may change this agreement at any time and it is your responsibility to review the most updated version of it on PGR's website at <http://www.ptgrey.com/support/kb/data/PGR-FlyCap-SDK-LA.pdf>. By continuing to use the SDK following such changes, you agree to be bound by them.

1. **Grant of License:** Subject to the terms of this agreement, you are hereby granted a limited, terminable, non-transferable, non-exclusive license and right to use the SDK only in conjunction with: (a) those PGR cameras listed at <http://www.ptgrey.com/products/index.asp> (as such list may be amended by PGR at any time and from time to time) and owned by you; and (b) the images derived from such cameras.
2. **Free and Open Source Components:** Notwithstanding anything to the contrary herein, use, copying and distribution of components of the SDK licensed under free and open source license agreements are governed solely by the terms of those license agreements (which are contained in the electronic documentation for the SDK) and not this agreement.
3. **Restrictions:** Except as (and only if) explicitly permitted by Section 4 below, you will not, and will not permit any third parties to: (a) copy the SDK, other than a reasonable number of backup copies for your own use only, and such backup copies together with the original will be kept in your possession and control; (b) provide or disclose the SDK to any third party; (c) alter, modify, reverse engineer, decompile or disassemble the SDK, or attempt to do any of the foregoing; (d) grant sublicenses, leases, or any other rights in the SDK to any third party; or (e) remove, alter or obscure any proprietary rights notices (including any copyright and trademark notices) on and in the SDK.
4. **Additional OEM Rights:** If you are an original equipment manufacturer, then in addition to the rights set out in Section 1 above you are hereby granted a limited, terminable, non-transferable, non-exclusive license and right to use the SDK for the sole additional purpose of incorporating the libraries found in the SDK (collectively, the "**Libraries**") into new products developed by you, in whole or in part, using the SDK (collectively, the "**Derivative Products**") provided that you: (a) ensure that the components of any Derivative Product that derive functionality from any of the Libraries may only be used with PGR products, including the SDK, and images derived from such products; (b) may only redistribute drivers (.inf and .sys), dynamically linked libraries (.dlls and .so), executables (.exe) and documentation (.doc, .txt, .pdf and .chm) and only to the extent necessary to support your Derivative Products. For clarity, headers (.h), source (.c, .cpp, .cs and .vb) and statically linked libraries (.lib and .a) cannot be redistributed; (c) will prohibit any, and ensure that there is no, redistribution of any of the Libraries by any third party, including any end user customers; and (d) will include any PGR and third party proprietary rights legends or notices (including copyright and trademark notices), unaltered and unobscured, on all Derivative Products.
5. **Ownership:** PGR and third parties are the owners of and retain title to all proprietary and intellectual property rights (including all patent, copyright, trade secret and trademark

rights) in and to the SDK. You have no right, title or interest in the SDK, except as specifically set forth herein, and no rights in any trade-marks of PGR. All rights not explicitly granted herein are hereby reserved.

6. **Indemnification:** You assume the entire risk relating to, and will indemnify, hold harmless and defend PGR from and against any claims, actions, lawsuits, or proceedings, and any losses, liabilities, damages and expenses (including attorney's fees and expenses) that arise or result from your activities under this agreement, including the distribution or use of the SDK (including the Libraries) and/or the development, distribution or use of any Derivative Product (including any intellectual property infringement claims relating thereto).

7. **No Warranties:** Your use of the SDK is solely at your own risk. The SDK is provided "as is" and "as available" without warranty or condition of any kind, either express, implied or statutory, including implied warranties of merchantability, fitness for a particular application or purpose, title and non-infringement, and PGR hereby expressly disclaims all such warranties and conditions. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

8. **Limitation of Liability:** Notwithstanding any other provision herein, to the maximum extent permitted by applicable law, PGR will not under any circumstances be liable for any direct, indirect, incidental, consequential, punitive or special losses or damages (including damages for bodily injury or death, business interruption, loss or corruption of business information or data, loss of opportunity, loss of privacy, cost of recovery, financial loss, loss of revenue or loss of profits for any reason whatsoever) whether or not PGR has been advised or should have been aware of the possibility of any such losses or damage arising, and in no event will the aggregate and cumulative liability of PGR for any matters arising under this agreement or otherwise exceed \$25.

9. **Changes and Support:** PGR may at its sole discretion elect to provide you with limited support services related to the SDK for such period of time as PGR at its sole discretion elects. PGR may change or cease to provide the SDK and the limited Support at any time and from time to time without notice to you and that PGR is not obligated to provide the SDK or any support. PGR may use any technical information, feedback or ideas you provide to PGR for PGR's business purposes, including product support and development.

10. **Termination:** PGR may terminate this agreement immediately on notice to you if you violate any of the terms of this agreement. PGR may also terminate this agreement for convenience provided that 30 days notice is delivered to you. Any notice given by PGR hereunder will be effective when sent by PGR to the email address you provided to PGR when you registered for a download account. On termination, for any reason, your rights hereunder will cease and you must immediately return all copies of the SDK to PGR and on termination for violation or default, you will be liable to PGR for all damages suffered as a result of the violation or default. Sections 6-8, 10 and 12 will survive any termination hereof.

11. **Export Laws:** This agreement involves products and/or technical data that may be controlled under laws and regulations of the United States and other countries, including the United States Export Administration Regulations, or any other applicable law, regulation, rule, guideline or order (collectively, "**Export Laws**"). You will comply with all Export Laws to ensure that the SDK is not exported, directly or indirectly, in contravention of the Export Laws.

You represent and warrant to PGR that you are not a person barred from receiving the SDK under any Export Laws.

12. **General:** This agreement is the entire agreement between you and PGR with respect to the subject matter of this agreement. If you are signing on behalf of an entity such as a corporation, you represent and warrant that you have the authority to bind such entity. This agreement and the rights granted hereunder are personal to you and you may not assign this agreement to a third party without the prior written consent of PGR. This agreement is governed exclusively by and will be enforced, construed, and interpreted exclusively in accordance with the laws of British Columbia ("BC") and the laws of Canada applicable in BC. The courts of the Province of BC will have exclusive jurisdiction over any dispute arising under this agreement. You agree that termination and/or monetary damages may not be a sufficient remedy if you breach this agreement and that PGR will be entitled, without waiving any other rights or remedies, to injunctive or equitable relief as may be deemed proper by a court of competent jurisdiction in the event of a breach. If PGR does not exercise any legal right or remedy in this agreement or otherwise, this will not be taken to be a formal waiver by PGR of its rights, which rights will remain available to PGR. If any provision of this agreement is construed to be illegal or invalid, the illegal or invalid provisions will be deemed stricken and deleted herefrom to the same extent and effect as if never incorporated herein, but all other provisions hereof will continue in full force and effect.

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Class Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	Module Documentation	11
5.1	Enumerations	11
5.1.1	Enumeration Type Documentation	13
5.1.1.1	BandwidthAllocation	13
5.1.1.2	BayerTileFormat	13
5.1.1.3	BusSpeed	14
5.1.1.4	ByteOrder	14
5.1.1.5	ColorProcessingAlgorithm	14
5.1.1.6	DriverType	15
5.1.1.7	ErrorType	15
5.1.1.8	FrameRate	17
5.1.1.9	GigEPropertyType	17
5.1.1.10	GrabMode	17
5.1.1.11	GrabTimeout	18
5.1.1.12	ImageFileFormat	18

5.1.1.13	InterfaceType	19
5.1.1.14	ManagedCallbackType	19
5.1.1.15	Mode	19
5.1.1.16	OSType	20
5.1.1.17	PCleBusSpeed	21
5.1.1.18	PixelFormat	21
5.1.1.19	PropertyType	22
5.1.1.20	StatisticsChannel	22
5.1.1.21	VideoMode	23
5.2	Structures	24
5.3	Image saving structures.	26
5.3.1	Detailed Description	26
6	Namespace Documentation	27
6.1	FlyCapture2 Namespace Reference	27
6.2	FlyCapture2Managed Namespace Reference	27
6.2.1	Function Documentation	32
6.2.1.1	EnumCallback	32
6.2.1.2	ImageCallbackDelegate	32
6.2.1.3	ImageEventCallback	32
6.3	FlyCapture2Managed::Gui Namespace Reference	32
7	Class Documentation	33
7.1	AviOption Struct Reference	33
7.1.1	Detailed Description	33
7.1.2	Constructor & Destructor Documentation	33
7.1.2.1	AviOption	33
7.1.3	Property Documentation	33
7.1.3.1	frameRate	33
7.2	CameraControlDialog Class Reference	34
7.2.1	Detailed Description	34
7.2.2	Constructor & Destructor Documentation	34
7.2.2.1	CameraControlDialog	34
7.2.2.2	~CameraControlDialog	34
7.2.3	Member Function Documentation	34

7.2.3.1	Connect	34
7.2.3.2	Disconnect	34
7.2.3.3	Hide	34
7.2.3.4	IsVisible	34
7.2.3.5	SetTitle	34
7.2.3.6	Show	34
7.3	CameraInfo Struct Reference	34
7.3.1	Detailed Description	36
7.3.2	Property Documentation	36
7.3.2.1	applicationIPAddress	36
7.3.2.2	applicationPort	36
7.3.2.3	bayerTileFormat	36
7.3.2.4	busNumber	37
7.3.2.5	ccpStatus	37
7.3.2.6	configROM	37
7.3.2.7	defaultGateway	37
7.3.2.8	driverName	37
7.3.2.9	driverType	37
7.3.2.10	firmwareBuildTime	37
7.3.2.11	firmwareVersion	37
7.3.2.12	gigEMajorVersion	37
7.3.2.13	gigEMinorVersion	37
7.3.2.14	iidcVersion	38
7.3.2.15	interfaceType	38
7.3.2.16	ipAddress	38
7.3.2.17	isColorCamera	38
7.3.2.18	macAddress	38
7.3.2.19	maximumBusSpeed	38
7.3.2.20	modelName	38
7.3.2.21	nodeNumber	38
7.3.2.22	pcieBusSpeed	38
7.3.2.23	sensorInfo	38
7.3.2.24	sensorResolution	39
7.3.2.25	serialNumber	39

7.3.2.26	subnetMask	39
7.3.2.27	userDefinedName	39
7.3.2.28	vendorName	39
7.3.2.29	xmlURL1	39
7.3.2.30	xmlURL2	39
7.4	CameraProperty Struct Reference	39
7.4.1	Detailed Description	40
7.4.2	Constructor & Destructor Documentation	40
7.4.2.1	CameraProperty	40
7.4.2.2	CameraProperty	40
7.4.3	Property Documentation	40
7.4.3.1	absControl	40
7.4.3.2	absValue	40
7.4.3.3	autoManualMode	40
7.4.3.4	onePush	41
7.4.3.5	onOff	41
7.4.3.6	present	41
7.4.3.7	type	41
7.4.3.8	valueA	41
7.4.3.9	valueB	41
7.5	CameraPropertyInfo Struct Reference	41
7.5.1	Detailed Description	42
7.5.2	Constructor & Destructor Documentation	42
7.5.2.1	CameraPropertyInfo	42
7.5.2.2	CameraPropertyInfo	42
7.5.3	Property Documentation	42
7.5.3.1	absMax	42
7.5.3.2	absMin	42
7.5.3.3	absValSupported	43
7.5.3.4	autoSupported	43
7.5.3.5	manualSupported	43
7.5.3.6	max	43
7.5.3.7	min	43
7.5.3.8	onePushSupported	43

7.5.3.9	onOffSupported	43
7.5.3.10	present	43
7.5.3.11	readOutSupported	43
7.5.3.12	type	43
7.5.3.13	unitAbbr	44
7.5.3.14	units	44
7.6	CameraSelectionDialog Class Reference	44
7.6.1	Detailed Description	44
7.6.2	Constructor & Destructor Documentation	44
7.6.2.1	CameraSelectionDialog	44
7.6.2.2	~CameraSelectionDialog	44
7.6.3	Member Function Documentation	44
7.6.3.1	GetSelectedCameraGuids	44
7.6.3.2	SetTitle	45
7.6.3.3	ShowModal	45
7.7	ConfigROM Struct Reference	45
7.7.1	Detailed Description	46
7.7.2	Property Documentation	46
7.7.2.1	chipIdHi	46
7.7.2.2	chipIdLo	46
7.7.2.3	keyword	46
7.7.2.4	nodeVendorId	46
7.7.2.5	unitSpecId	46
7.7.2.6	unitSubSWVer	46
7.7.2.7	unitSWVer	46
7.7.2.8	vendorUniqueInfo0	46
7.7.2.9	vendorUniqueInfo1	46
7.7.2.10	vendorUniqueInfo2	47
7.7.2.11	vendorUniqueInfo3	47
7.8	EmbeddedImageInfo Struct Reference	47
7.8.1	Detailed Description	47
7.8.2	Constructor & Destructor Documentation	47
7.8.2.1	EmbeddedImageInfo	47
7.8.3	Property Documentation	47

7.8.3.1	brightness	47
7.8.3.2	exposure	47
7.8.3.3	frameCounter	48
7.8.3.4	gain	48
7.8.3.5	GPIOPinState	48
7.8.3.6	ROIPosition	48
7.8.3.7	shutter	48
7.8.3.8	strobePattern	48
7.8.3.9	timestamp	48
7.8.3.10	whiteBalance	48
7.9	EmbeddedImageInfoProperty Struct Reference	48
7.9.1	Detailed Description	48
7.9.2	Property Documentation	48
7.9.2.1	available	48
7.9.2.2	onOff	48
7.10	FC2Config Struct Reference	49
7.10.1	Detailed Description	49
7.10.2	Constructor & Destructor Documentation	50
7.10.2.1	FC2Config	50
7.10.3	Property Documentation	50
7.10.3.1	asyncBusSpeed	50
7.10.3.2	bandwidthAllocation	50
7.10.3.3	grabMode	50
7.10.3.4	grabTimeout	50
7.10.3.5	isochBusSpeed	50
7.10.3.6	minNumImageNotifications	50
7.10.3.7	numBuffers	50
7.10.3.8	numImageNotifications	50
7.10.3.9	registerTimeout	51
7.10.3.10	registerTimeoutRetries	51
7.11	FC2Exception Class Reference	51
7.11.1	Detailed Description	52
7.11.2	Constructor & Destructor Documentation	52
7.11.2.1	FC2Exception	52

7.11.2.2	FC2Exception	52
7.11.2.3	FC2Exception	52
7.11.2.4	~FC2Exception	52
7.11.2.5	FC2Exception	52
7.11.2.6	FC2Exception	52
7.11.3	Property Documentation	52
7.11.3.1	CauseType	52
7.11.3.2	NativeErrorTrace	52
7.11.3.3	Type	52
7.12	FC2Version Struct Reference	52
7.12.1	Detailed Description	53
7.12.2	Property Documentation	53
7.12.2.1	build	53
7.12.2.2	major	53
7.12.2.3	minor	53
7.12.2.4	type	53
7.13	Format7ImageSettings Struct Reference	53
7.13.1	Detailed Description	54
7.13.2	Property Documentation	54
7.13.2.1	height	54
7.13.2.2	mode	54
7.13.2.3	offsetX	54
7.13.2.4	offsetY	54
7.13.2.5	pixelFormat	54
7.13.2.6	width	54
7.14	Format7Info Struct Reference	54
7.14.1	Detailed Description	55
7.14.2	Property Documentation	55
7.14.2.1	imageHStepSize	55
7.14.2.2	imageVStepSize	55
7.14.2.3	maxHeight	56
7.14.2.4	maxPacketSize	56
7.14.2.5	maxWidth	56
7.14.2.6	minPacketSize	56

7.14.2.7	mode	56
7.14.2.8	offsetHStepSize	56
7.14.2.9	offsetVStepSize	56
7.14.2.10	packetSize	56
7.14.2.11	percentage	56
7.14.2.12	pixelFormatBitField	56
7.14.2.13	vendorPixelFormatBitField	57
7.15	Format7PacketInfo Struct Reference	57
7.15.1	Detailed Description	57
7.15.2	Property Documentation	57
7.15.2.1	maxBytesPerPacket	57
7.15.2.2	recommendedBytesPerPacket	57
7.15.2.3	unitBytesPerPacket	57
7.16	GigEConfig Struct Reference	57
7.16.1	Detailed Description	58
7.16.2	Property Documentation	58
7.16.2.1	enablePacketResend	58
7.17	GigEImageSettings Struct Reference	58
7.17.1	Detailed Description	58
7.17.2	Property Documentation	59
7.17.2.1	height	59
7.17.2.2	offsetX	59
7.17.2.3	offsetY	59
7.17.2.4	pixelFormat	59
7.17.2.5	width	59
7.18	GigEImageSettingsInfo Struct Reference	59
7.18.1	Detailed Description	60
7.18.2	Property Documentation	60
7.18.2.1	imageHStepSize	60
7.18.2.2	imageVStepSize	60
7.18.2.3	maxHeight	60
7.18.2.4	maxWidth	60
7.18.2.5	offsetHStepSize	60
7.18.2.6	offsetVStepSize	60

7.18.2.7	pixelFormatBitField	60
7.18.2.8	vendorPixelFormatBitField	60
7.19	GigEProperty Struct Reference	61
7.19.1	Detailed Description	61
7.19.2	Property Documentation	61
7.19.2.1	isReadable	61
7.19.2.2	isWritable	61
7.19.2.3	max	61
7.19.2.4	min	61
7.19.2.5	propType	62
7.19.2.6	value	62
7.20	GigEStreamChannel Struct Reference	62
7.20.1	Detailed Description	62
7.20.2	Property Documentation	62
7.20.2.1	destinationIpAddress	62
7.20.2.2	doNotFragment	63
7.20.2.3	hostPort	63
7.20.2.4	interPacketDelay	63
7.20.2.5	networkInterfaceIndex	63
7.20.2.6	packetSize	63
7.20.2.7	sourcePort	63
7.21	H264Option Struct Reference	63
7.21.1	Detailed Description	64
7.21.2	Constructor & Destructor Documentation	64
7.21.2.1	H264Option	64
7.21.3	Property Documentation	64
7.21.3.1	bitrate	64
7.21.3.2	frameRate	64
7.21.3.3	height	64
7.21.3.4	width	64
7.22	ImageMetadata Struct Reference	64
7.22.1	Detailed Description	65
7.22.2	Property Documentation	65
7.22.2.1	embeddedBrightness	65

7.22.2.2	embeddedExposure	65
7.22.2.3	embeddedFrameCounter	65
7.22.2.4	embeddedGain	65
7.22.2.5	embeddedGPIOPinState	65
7.22.2.6	embeddedROIPosition	66
7.22.2.7	embeddedShutter	66
7.22.2.8	embeddedStrobePattern	66
7.22.2.9	embeddedTimeStamp	66
7.22.2.10	embeddedWhiteBalance	66
7.23	JpegOption Struct Reference	66
7.23.1	Detailed Description	66
7.23.2	Constructor & Destructor Documentation	67
7.23.2.1	JpegOption	67
7.23.3	Property Documentation	67
7.23.3.1	progressive	67
7.23.3.2	quality	67
7.24	Jpg2Option Struct Reference	67
7.24.1	Detailed Description	67
7.24.2	Constructor & Destructor Documentation	67
7.24.2.1	Jpg2Option	67
7.24.3	Property Documentation	68
7.24.3.1	quality	68
7.25	LutData Struct Reference	68
7.25.1	Detailed Description	68
7.25.2	Property Documentation	68
7.25.2.1	enabled	68
7.25.2.2	inputBitDepth	68
7.25.2.3	numBanks	69
7.25.2.4	numChannels	69
7.25.2.5	numEntries	69
7.25.2.6	outputBitDepth	69
7.25.2.7	supported	69
7.26	ManagedAVIRecorder Class Reference	69
7.26.1	Detailed Description	70

7.26.2	Constructor & Destructor Documentation	70
7.26.2.1	ManagedAVIRecorder	70
7.26.2.2	~ManagedAVIRecorder	70
7.26.3	Member Function Documentation	70
7.26.3.1	AVIAppend	70
7.26.3.2	AVIClose	70
7.26.3.3	AVIOpen	70
7.26.3.4	AVIOpen	71
7.26.3.5	AVIOpen	71
7.27	ManagedBusManager Class Reference	71
7.27.1	Detailed Description	73
7.27.2	Constructor & Destructor Documentation	73
7.27.2.1	ManagedBusManager	73
7.27.2.2	~ManagedBusManager	73
7.27.2.3	!ManagedBusManager	74
7.27.3	Member Function Documentation	74
7.27.3.1	ConvertToManagedGuid	74
7.27.3.2	ConvertToNativeGuid	74
7.27.3.3	DiscoverGigECameras	74
7.27.3.4	FireBusReset	74
7.27.3.5	ForceAllIPAddressesAutomatically	75
7.27.3.6	ForceAllIPAddressesAutomatically	75
7.27.3.7	ForceIPAddressToCamera	75
7.27.3.8	GetCameraFromIndex	75
7.27.3.9	GetCameraFromIPAddress	76
7.27.3.10	GetCameraFromSerialNumber	76
7.27.3.11	GetCameraSerialNumberFromIndex	76
7.27.3.12	GetDeviceFromIndex	77
7.27.3.13	GetInterfaceTypeFromGuid	77
7.27.3.14	GetNumOfCameras	77
7.27.3.15	GetNumOfDevices	78
7.27.3.16	GetTopology	78
7.27.3.17	GetUsbLinkInfo	78
7.27.3.18	GetUsbPortStatus	78

7.27.3.19	IsCameraControlable	78
7.27.3.20	ReadPhyRegister	79
7.27.3.21	RegisterCallback	79
7.27.3.22	RescanBus	80
7.27.3.23	UnregisterCallback	80
7.27.3.24	WritePhyRegister	80
7.28	ManagedCamera Class Reference	80
7.28.1	Detailed Description	83
7.28.2	Constructor & Destructor Documentation	83
7.28.2.1	ManagedCamera	83
7.28.2.2	~ManagedCamera	83
7.28.2.3	IManagedCamera	83
7.28.3	Member Function Documentation	83
7.28.3.1	Connect	83
7.28.3.2	GetFormat7Configuration	83
7.28.3.3	GetFormat7Info	84
7.28.3.4	GetVideoModeAndFrameRate	84
7.28.3.5	GetVideoModeAndFrameRateInfo	85
7.28.3.6	SetFormat7Configuration	85
7.28.3.7	SetFormat7Configuration	85
7.28.3.8	SetVideoModeAndFrameRate	86
7.28.3.9	StartSyncCapture	86
7.28.3.10	StartSyncCapture	87
7.28.3.11	ValidateFormat7Settings	87
7.29	ManagedCameraBase Class Reference	88
7.29.1	Detailed Description	92
7.29.2	Constructor & Destructor Documentation	92
7.29.2.1	~ManagedCameraBase	92
7.29.2.2	ManagedCameraBase	92
7.29.3	Member Function Documentation	92
7.29.3.1	Connect	92
7.29.3.2	Disconnect	92
7.29.3.3	EnableLUT	92
7.29.3.4	FireSoftwareTrigger	93

7.29.3.5	GetActiveLUTBank	93
7.29.3.6	GetCameraInfo	93
7.29.3.7	GetConfiguration	93
7.29.3.8	GetCycleTime	94
7.29.3.9	GetEmbeddedImageInfo	94
7.29.3.10	GetGPIOPinDirection	94
7.29.3.11	GetLUTBankInfo	95
7.29.3.12	GetLUTChannel	95
7.29.3.13	GetLUTInfo	95
7.29.3.14	GetMemoryChannel	96
7.29.3.15	GetMemoryChannelInfo	96
7.29.3.16	GetNativeCamera	96
7.29.3.17	GetProperty	96
7.29.3.18	GetPropertyInfo	97
7.29.3.19	GetRegisterString	97
7.29.3.20	GetStrobe	97
7.29.3.21	GetStrobeInfo	98
7.29.3.22	GetTriggerMode	98
7.29.3.23	GetTriggerModeInfo	98
7.29.3.24	IsConnected	99
7.29.3.25	OnNativeCallback	99
7.29.3.26	ReadRegister	99
7.29.3.27	ReadRegisterBlock	99
7.29.3.28	RestoreFromMemoryChannel	100
7.29.3.29	RetrieveBuffer	100
7.29.3.30	SaveToMemoryChannel	100
7.29.3.31	SetActiveLUTBank	101
7.29.3.32	SetCallback	101
7.29.3.33	SetCamera	101
7.29.3.34	SetConfiguration	101
7.29.3.35	SetEmbeddedImageInfo	101
7.29.3.36	SetGPIOPinDirection	102
7.29.3.37	SetGPIOPinDirection	102
7.29.3.38	SetLUTChannel	103

7.29.3.39 SetProperty	103
7.29.3.40 SetProperty	103
7.29.3.41 SetStrobe	104
7.29.3.42 SetTriggerMode	104
7.29.3.43 StartCapture	104
7.29.3.44 StartCapture	105
7.29.3.45 StopCapture	105
7.29.3.46 WaitForBufferEvent	105
7.29.3.47 WriteRegister	106
7.29.3.48 WriteRegister	106
7.29.3.49 WriteRegisterBlock	106
7.29.4 Member Data Documentation	106
7.29.4.1 m_externalDelegate	106
7.29.4.2 m_internalDelegate	107
7.29.4.3 m_isLocal	107
7.29.4.4 m_pNativeCamBase	107
7.30 ManagedGCCamera Class Reference	107
7.30.1 Constructor & Destructor Documentation	108
7.30.1.1 ManagedGCCamera	108
7.30.1.2 ~ManagedGCCamera	108
7.30.1.3 IManagedGCCamera	108
7.30.2 Member Function Documentation	108
7.30.2.1 Connect	108
7.30.2.2 Connect	108
7.30.2.3 Disconnect	108
7.30.2.4 GetNodeMap	109
7.30.2.5 SetCamera	109
7.30.2.6 SetCamera	109
7.31 ManagedGCPort Class Reference	109
7.31.1 Constructor & Destructor Documentation	109
7.31.1.1 ManagedGCPort	109
7.31.1.2 ~ManagedGCPort	109
7.31.2 Member Function Documentation	109
7.31.2.1 Read	109

7.31.2.2	Write	109
7.32	ManagedGigECamera Class Reference	109
7.32.1	Detailed Description	113
7.32.2	Constructor & Destructor Documentation	113
7.32.2.1	ManagedGigECamera	113
7.32.2.2	~ManagedGigECamera	113
7.32.2.3	!ManagedGigECamera	113
7.32.3	Member Function Documentation	113
7.32.3.1	Connect	113
7.32.3.2	DiscoverGigEPacketSize	113
7.32.3.3	GetGigEConfig	114
7.32.3.4	GetGigEImageBinningSettings	114
7.32.3.5	GetGigEImageSettings	114
7.32.3.6	GetGigEImageSettingsInfo	114
7.32.3.7	GetGigEImagingMode	114
7.32.3.8	GetGigEProperty	115
7.32.3.9	GetGigEStreamChannelInfo	115
7.32.3.10	GetNumStreamChannels	115
7.32.3.11	QueryGigEImagingMode	115
7.32.3.12	ReadGVCPMemory	115
7.32.3.13	ReadGVCPRegister	116
7.32.3.14	ReadGVCPRegisterBlock	116
7.32.3.15	SetGigEConfig	116
7.32.3.16	SetGigEImageBinningSettings	116
7.32.3.17	SetGigEImageSettings	117
7.32.3.18	SetGigEImagingMode	117
7.32.3.19	SetGigEProperty	117
7.32.3.20	SetGigEStreamChannelInfo	117
7.32.3.21	WriteGVCPMemory	118
7.32.3.22	WriteGVCPRegister	118
7.32.3.23	WriteGVCPRegister	118
7.32.3.24	WriteGVCPRegisterBlock	118
7.33	ManagedImage Class Reference	118
7.33.1	Detailed Description	121

7.33.2	Constructor & Destructor Documentation	121
7.33.2.1	ManagedImage	121
7.33.2.2	ManagedImage	121
7.33.2.3	ManagedImage	121
7.33.2.4	ManagedImage	121
7.33.2.5	ManagedImage	121
7.33.2.6	ManagedImage	121
7.33.2.7	ManagedImage	121
7.33.2.8	ManagedImage	121
7.33.2.9	ManagedImage	122
7.33.2.10	~ManagedImage	122
7.33.2.11	ManagedImage	122
7.33.2.12	IManagedImage	122
7.33.3	Member Function Documentation	122
7.33.3.1	CalculateStatistics	122
7.33.3.2	Convert	122
7.33.3.3	Convert	122
7.33.3.4	DetermineBitsPerPixel	123
7.33.3.5	GetNativeImage	123
7.33.3.6	GetRawNativeImagePointer	123
7.33.3.7	IsNativeImageValid	123
7.33.3.8	ReleaseBuffer	123
7.33.3.9	Save	123
7.33.3.10	Save	123
7.33.3.11	Save	123
7.33.3.12	Save	124
7.33.3.13	Save	124
7.33.3.14	Save	124
7.33.3.15	Save	124
7.33.3.16	Save	124
7.33.3.17	SetData	125
7.33.3.18	SetDimensions	125
7.33.4	Property Documentation	125
7.33.4.1	bayerTileFormat	125

7.33.4.2	bitmap	125
7.33.4.3	bitsPerPixel	126
7.33.4.4	colorProcessingAlgorithm	126
7.33.4.5	cols	126
7.33.4.6	data	126
7.33.4.7	defaultColorProcessingAlgorithm	126
7.33.4.8	defaultOutputPixelFormat	126
7.33.4.9	imageMetadata	126
7.33.4.10	pixelFormat	126
7.33.4.11	receivedDataSize	126
7.33.4.12	rows	127
7.33.4.13	stride	127
7.33.4.14	timeStamp	127
7.34	ManagedImageStatistics Class Reference	127
7.34.1	Constructor & Destructor Documentation	128
7.34.1.1	ManagedImageStatistics	128
7.34.1.2	~ManagedImageStatistics	128
7.34.2	Member Function Documentation	128
7.34.2.1	DisableAll	128
7.34.2.2	EnableAll	128
7.34.2.3	EnableGreyOnly	128
7.34.2.4	EnableHSLOnly	128
7.34.2.5	EnableRGBOnly	128
7.34.2.6	GetChannelStatus	128
7.34.2.7	GetHistogram	128
7.34.2.8	GetMean	128
7.34.2.9	GetNativeImageStatistics	128
7.34.2.10	GetNumPixelValues	128
7.34.2.11	GetPixelValueRange	128
7.34.2.12	GetRange	128
7.34.2.13	GetStatistics	128
7.34.2.14	SetChannelStatus	128
7.35	ManagedPGRGuid Class Reference	128
7.35.1	Detailed Description	129

7.35.2	Constructor & Destructor Documentation	129
7.35.2.1	ManagedPGRGuid	129
7.35.2.2	ManagedPGRGuid	129
7.35.2.3	ManagedPGRGuid	130
7.35.3	Member Function Documentation	130
7.35.3.1	Equals	130
7.35.3.2	GetHashCode	130
7.35.3.3	operator!=	130
7.35.3.4	operator=	130
7.35.3.5	operator==	130
7.35.4	Member Data Documentation	130
7.35.4.1	value0	130
7.35.4.2	value1	130
7.35.4.3	value2	130
7.35.4.4	value3	130
7.36	ManagedTopologyNode Class Reference	130
7.36.1	Detailed Description	132
7.36.2	Member Enumeration Documentation	132
7.36.2.1	NodeType	132
7.36.2.2	PortType	132
7.36.3	Constructor & Destructor Documentation	132
7.36.3.1	~ManagedTopologyNode	132
7.36.3.2	ManagedTopologyNode	132
7.36.3.3	ManagedTopologyNode	132
7.36.3.4	ManagedTopologyNode	132
7.36.4	Member Function Documentation	132
7.36.4.1	GetChild	132
7.36.4.2	GetDeviceId	133
7.36.4.3	GetGuid	133
7.36.4.4	GetInterfaceType	133
7.36.4.5	GetNodeType	133
7.36.4.6	GetNumChildren	133
7.36.4.7	GetNumPorts	134
7.36.4.8	GetPortType	134

7.36.4.9	TranslateNodeType	134
7.36.4.10	TranslateNodeType	134
7.36.4.11	TranslatePortType	134
7.36.4.12	TranslatePortType	134
7.37	ManagedUtilities Class Reference	134
7.37.1	Member Function Documentation	135
7.37.1.1	LaunchBrowser	135
7.37.1.2	LaunchCommand	135
7.37.1.3	LaunchHelp	135
7.37.2	Property Documentation	135
7.37.2.1	libraryVersion	135
7.37.2.2	systemInfo	135
7.38	MJPGOption Struct Reference	135
7.38.1	Detailed Description	135
7.38.2	Constructor & Destructor Documentation	135
7.38.2.1	MJPGOption	135
7.38.3	Property Documentation	135
7.38.3.1	frameRate	135
7.38.3.2	quality	136
7.39	PgmOption Struct Reference	136
7.39.1	Detailed Description	136
7.39.2	Constructor & Destructor Documentation	136
7.39.2.1	PgmOption	136
7.39.3	Property Documentation	136
7.39.3.1	binaryFile	136
7.40	PngOption Struct Reference	136
7.40.1	Detailed Description	137
7.40.2	Constructor & Destructor Documentation	137
7.40.2.1	PngOption	137
7.40.3	Property Documentation	137
7.40.3.1	compressionLevel	137
7.40.3.2	interlaced	137
7.41	PpmOption Struct Reference	137
7.41.1	Detailed Description	137

7.41.2	Constructor & Destructor Documentation	138
7.41.2.1	PpmOption	138
7.41.3	Property Documentation	138
7.41.3.1	binaryFile	138
7.42	StrobeControl Struct Reference	138
7.42.1	Detailed Description	138
7.42.2	Property Documentation	138
7.42.2.1	delay	138
7.42.2.2	duration	138
7.42.2.3	onOff	139
7.42.2.4	polarity	139
7.42.2.5	source	139
7.43	StrobeInfo Struct Reference	139
7.43.1	Detailed Description	139
7.43.2	Property Documentation	139
7.43.2.1	maxValue	139
7.43.2.2	minValue	140
7.43.2.3	onOffSupported	140
7.43.2.4	polaritySupported	140
7.43.2.5	present	140
7.43.2.6	readOutSupported	140
7.43.2.7	source	140
7.44	SystemInfo Struct Reference	140
7.44.1	Detailed Description	141
7.44.2	Property Documentation	141
7.44.2.1	byteOrder	141
7.44.2.2	cpuDescription	141
7.44.2.3	driverList	141
7.44.2.4	gpuDescription	141
7.44.2.5	libraryList	141
7.44.2.6	numCpuCores	141
7.44.2.7	osDescription	142
7.44.2.8	osType	142
7.44.2.9	screenHeight	142

7.44.2.10	screenWidth	142
7.44.2.11	systemMemorySize	142
7.45	TiffOption Struct Reference	142
7.45.1	Detailed Description	142
7.45.2	Member Enumeration Documentation	143
7.45.2.1	CompressionMethod	143
7.45.3	Constructor & Destructor Documentation	143
7.45.3.1	TiffOption	143
7.45.4	Property Documentation	143
7.45.4.1	compression	143
7.46	TimeStamp Struct Reference	143
7.46.1	Detailed Description	144
7.46.2	Property Documentation	144
7.46.2.1	cycleCount	144
7.46.2.2	cycleOffset	144
7.46.2.3	cycleSeconds	144
7.46.2.4	microSeconds	144
7.46.2.5	seconds	144
7.47	Translate Class Reference	144
7.47.1	Member Function Documentation	147
7.47.1.1	ToMgd	147
7.47.1.2	ToMgd	147
7.47.1.3	ToMgd	147
7.47.1.4	ToMgd	147
7.47.1.5	ToMgd	148
7.47.1.6	ToMgd	148
7.47.1.7	ToMgd	148
7.47.1.8	ToMgd	148
7.47.1.9	ToMgd	148
7.47.1.10	ToMgd	148
7.47.1.11	ToMgd	148
7.47.1.12	ToMgd	148
7.47.1.13	ToMgd	148
7.47.1.14	ToMgd	148

7.47.1.15 ToMgd	148
7.47.1.16 ToMgd	148
7.47.1.17 ToMgd	148
7.47.1.18 ToMgd	148
7.47.1.19 ToMgd	148
7.47.1.20 ToMgd	148
7.47.1.21 ToMgd	149
7.47.1.22 ToMgd	149
7.47.1.23 ToMgd	149
7.47.1.24 ToMgd	149
7.47.1.25 ToMgd	149
7.47.1.26 ToMgd	149
7.47.1.27 ToMgd	149
7.47.1.28 ToMgd	149
7.47.1.29 ToMgd	149
7.47.1.30 ToMgd	149
7.47.1.31 ToMgd	149
7.47.1.32 ToNative	149
7.47.1.33 ToNative	149
7.47.1.34 ToNative	149
7.47.1.35 ToNative	149
7.47.1.36 ToNative	149
7.47.1.37 ToNative	150
7.47.1.38 ToNative	150
7.47.1.39 ToNative	150
7.47.1.40 ToNative	150
7.47.1.41 ToNative	150
7.47.1.42 ToNative	150
7.47.1.43 ToNative	150
7.47.1.44 ToNative	150
7.47.1.45 ToNative	150
7.47.1.46 ToNative	150
7.47.1.47 ToNative	150
7.47.1.48 ToNative	150

7.47.1.49 ToNative	150
7.47.1.50 ToNative	150
7.47.1.51 ToNative	150
7.47.1.52 ToNative	150
7.47.1.53 ToNative	151
7.47.1.54 ToNative	151
7.47.1.55 ToNative	151
7.47.1.56 ToNative	151
7.47.1.57 ToNative	151
7.47.1.58 ToNative	151
7.47.1.59 ToNative	151
7.47.1.60 ToNative	151
7.47.1.61 ToNative	151
7.47.1.62 ToNative	151
7.47.1.63 ToNative	151
7.47.1.64 translate	151
7.47.1.65 translate	151
7.47.1.66 translate	151
7.47.1.67 translate	151
7.47.1.68 translate	151
7.47.1.69 translate	152
7.47.1.70 translate	152
7.47.1.71 translate	152
7.47.1.72 translate	152
7.47.1.73 translate	152
7.47.1.74 translate	152
7.47.1.75 translate	152
7.47.1.76 translate	152
7.47.1.77 translate	152
7.47.1.78 translate	152
7.47.1.79 translate	152
7.47.1.80 translate	152
7.47.1.81 translate	152
7.47.1.82 translate	152

7.47.1.83 translate	152
7.47.1.84 translate	152
7.47.1.85 translate	152
7.47.1.86 translate	153
7.47.1.87 translate	153
7.47.1.88 translate	153
7.47.1.89 translate	153
7.47.1.90 translate	153
7.47.1.91 translate	153
7.47.1.92 translate	153
7.47.1.93 translate	153
7.47.1.94 translate	153
7.47.1.95 translate	153
7.47.1.96 translate	153
7.47.1.97 translate	153
7.47.1.98 translate	153
7.47.1.99 translate	153
7.47.1.100translate	153
7.47.1.101translate	153
7.47.1.102translate	154
7.47.1.103translate	154
7.47.1.104Translate::ToMgd	154
7.47.1.105Translate::ToNative	154
7.48 TriggerMode Struct Reference	154
7.48.1 Detailed Description	154
7.48.2 Property Documentation	154
7.48.2.1 mode	154
7.48.2.2 onOff	154
7.48.2.3 parameter	155
7.48.2.4 polarity	155
7.48.2.5 source	155
7.49 TriggerModelInfo Struct Reference	155
7.49.1 Detailed Description	155
7.49.2 Property Documentation	156

7.49.2.1	modeMask	156
7.49.2.2	onOffSupported	156
7.49.2.3	polaritySupported	156
7.49.2.4	present	156
7.49.2.5	readOutSupported	156
7.49.2.6	softwareTriggerSupported	156
7.49.2.7	sourceMask	156
7.49.2.8	valueReadable	156

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Enumerations	11
Structures	24
Image saving structures.	26

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

FlyCapture2	27
FlyCapture2Managed	27
FlyCapture2Managed::Gui	32

Chapter 3

Class Index

3.1 Class Hierarchy

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

AviOption	33
CameraControlDialog	34
CameraInfo	34
CameraProperty	39
CameraPropertyInfo	41
CameraSelectionDialog	44
ConfigROM	45
EmbeddedImageInfo	47
EmbeddedImageInfoProperty	48
FC2Config	49
FC2Exception	51
FC2Version	52
Format7ImageSettings	53
Format7Info	54
Format7PacketInfo	57
GigEConfig	57
GigEImageSettings	58
GigEImageSettingsInfo	59
GigEProperty	61
GigEStreamChannel	62
H264Option	63
ImageMetadata	64
JpegOption	66
Jpg2Option	67
LutData	68
ManagedAVIRecorder	69
ManagedBusManager	71
ManagedCameraBase	88
ManagedCamera	80

ManagedGCCamera	107
ManagedGigECamera	109
ManagedGCPort	109
ManagedImage	118
ManagedImageStatistics	127
ManagedPGRGuid	128
ManagedTopologyNode	130
ManagedUtilities	134
MJPGOption	135
PgmOption	136
PngOption	136
PpmOption	137
StrobeControl	138
StrobeInfo	139
SystemInfo	140
TiffOption	142
TimeStamp	143
Translate	144
TriggerMode	154
TriggerModelInfo	155

Chapter 4

Class Index

4.1 Class List

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

AviOption	Options for saving AVI files	33
CameraControlDialog	CameraControlDialog : managed wrapper of FlyCapture2::CameraControlDialog (see for details)	34
CameraInfo	Camera information	34
CameraProperty	A specific camera property	39
CameraPropertyInfo	Information about a specific camera property	41
CameraSelectionDialog	CameraControlDialog : managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)	44
ConfigROM	Camera configuration ROM	45
EmbeddedImageInfo	Properties of the possible embedded image information	47
EmbeddedImageInfoProperty	Properties of a single embedded image info property	48
FC2Config	Configuration for a camera	49
FC2Exception	Exception that is thrown when an error is encountered	51
FC2Version	The current version of the library	52
Format7ImageSettings	Format 7 image settings	53

Format7Info	
Format 7 information for a single mode	54
Format7PacketInfo	
Format 7 packet information	57
GigEConfig	
Configuration for a GigE camera	57
GigEImageSettings	
Image settings for a GigE camera	58
GigEImageSettingsInfo	
Format 7 information for a single mode	59
GigEProperty	
A GigE property	61
GigEStreamChannel	
Information about a single GigE stream channel	62
H264Option	
Options for saving H.264 files	63
ImageMetadata	
Metadata related to an image	64
JpegOption	
Options for saving JPEG image	66
Jpg2Option	
Options for saving JPEG2000 image	67
LutData	
Information about the camera's look up table	68
ManagedAVIRecorder	
ManagedAVIRecorder provides the functionality for the user to record images to an AVI file	69
ManagedBusManager	
ManagedBusManager provides the functionality for the user to get an PGRGuid for a desired camera or device easily	71
ManagedCamera	
ManagedCamera represents a physical camera that uses the IIDC register set	80
ManagedCameraBase	
Abstract base class that represents a generic camera that defines a general interface to a camera	88
ManagedGCCamera	107
ManagedGCPort	109
ManagedGigECamera	
The GigECamera object represents a physical Gigabit Ethernet cam- era	109
ManagedImage	
The ManagedImageImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk	118
ManagedImageStatistics	127
ManagedPGRGuid	
Managed version of a PGRGuid	128

ManagedTopologyNode	
Topology information that can be used to generate a tree structure of all cameras and devices connected to a computer	130
ManagedUtilities	134
MJPEGOption	
Options for saving MJPEG files	135
PgmOption	
Options for saving PGM images	136
PngOption	
Options for saving PNG images	136
PpmOption	
Options for saving PPM images	137
StrobeControl	
A camera strobe	138
StrobeInfo	
A camera strobe property	139
SystemInfo	
Description of the system	140
TiffOption	
Options for saving TIFF images	142
TimeStamp	
Timestamp information	143
Translate	144
TriggerMode	
A camera trigger	154
TriggerModelInfo	
Information about a camera trigger property	155

Chapter 5

Module Documentation

5.1 Enumerations

Enumerations

- enum `ErrorType` { `Undefined` = -1, `Ok`, `Failed`, `NotImplemented`, `FailedBusMasterConnection`, `NotConnected`, `InitFailed`, `NotInitialized`, `InvalidParameter`, `InvalidSettings`, `InvalidBuManager`, `MemoryAllocationFailed`, `LowLevelFailure`, `NotFound`, `FailedGuid`, `InvalidPacketSize`, `InvalidMode`, `NotInFormat7`, `×` `NotSupported`, `Timeout`, `BusMasterFailed`, `InvalidGeneration`, `LutFailed`, `×` `lfdcFailed`, `StrobeFailed`, `TriggerFailed`, `PropertyFailed`, `PropertyNotPresent`, `RegisterFailed`, `ReadRegisterFailed`, `WriteRegisterFailed`, `IsochFailed`, `×` `IsochAlreadyStarted`, `IsochNotStarted`, `IsochStartFailed`, `IsochRetrieveBufferFailed`, `IsochStopFailed`, `IsochSyncFailed`, `IsochBandwidthExceeded`, `ImageConversionFailed`, `ImageLibraryFailure`, `BufferTooSmall`, `ImageConsistencyError` }

The error types returned by functions.

- enum `ManagedCallbackType` { `BusReset`, `Arrival`, `Removal` }

The type of bus callback to register a callback function for.

- enum `GrabMode` { `DropFrames`, `BufferFrames`, `Unspecified` = -2 }

The grab strategy employed during image transfer.

- enum `GrabTimeout` { `None` = 0, `Infinite` = -1, `Unspecified` = -2 }

Timeout options for grabbing images.

- enum `BandwidthAllocation` { `Off` = 0, `On` = 1, `Unsupported` = 2, `Unspecified` = -2 }

Bandwidth allocation options for 1394 devices.

- enum `InterfaceType` { `Ieee1394`, `Usb2`, `Usb3`, `GigE`, `Unknown` = -1 }

Interfaces that a camera may use to communicate with a host.

- enum `PropertyType` { `Brightness`, `AutoExposure`, `Sharpness`, `WhiteBalance`, `Hue`, `Saturation`, `Gamma`, `Iris`, `Focus`, `Zoom`, `Pan`, `Tilt`, `Shutter`, `Gain`, `TriggerMode`, `TriggerDelay`, `FrameRate`, `Temperature`, `Unspecified` = -2 }

Camera properties.

- enum `FrameRate` { `FrameRate1_875`, `FrameRate3_75`, `FrameRate7_5`, `FrameRate15`, `FrameRate30`, `FrameRate60`, `FrameRate120`, `FrameRate240`, `FrameRateFormat7`, `NumberOfFrameRates` }

Frame rates in frames per second.

- enum `VideoMode` { `VideoMode160x120Yuv444`, `VideoMode320x240Yuv422`, `VideoMode640x480Yuv411`, `VideoMode640x480Yuv422`, `VideoMode640x480Rbg`, `VideoMode640x480Y8`, `VideoMode640x480Y16`, `VideoMode800x600Yuv422`, `VideoMode800x600Rbg`, `VideoMode800x600Y8`, `VideoMode800x600Y16`, `VideoMode1024x768Yuv422`, `VideoMode1024x768Rbg`, `VideoMode1024x768Y8`, `VideoMode1024x768Y16`, `VideoMode1280x960Yuv422`, `VideoMode1280x960Rbg`, `VideoMode1280x960Y8`, `VideoMode1280x960Y16`, `VideoMode1600x1200Yuv422`, `VideoMode1600x1200Rbg`, `VideoMode1600x1200Y8`, `VideoMode1600x1200Y16`, `VideoModeFormat7`, `NumberOfVideoModes` }

DCAM video modes.

- enum `Mode` { `Mode0` = 0, `Mode1`, `Mode2`, `Mode3`, `Mode4`, `Mode5`, `Mode6`, `Mode7`, `Mode8`, `Mode9`, `Mode10`, `Mode11`, `Mode12`, `Mode13`, `Mode14`, `Mode15`, `Mode16`, `Mode17`, `Mode18`, `Mode19`, `Mode20`, `Mode21`, `Mode22`, `Mode23`, `Mode24`, `Mode25`, `Mode26`, `Mode27`, `Mode28`, `Mode29`, `Mode30`, `Mode31`, `NumberOfModes` }

Camera modes for DCAM formats as well as Format7.

- enum `PixelFormat` { `PixelFormatMono8` = 0x80000000, `PixelFormat411Yuv8` = 0x40000000, `PixelFormat422Yuv8` = 0x20000000, `PixelFormat444Yuv8` = 0x10000000, `PixelFormatRgb8` = 0x08000000, `PixelFormatMono16` = 0x04000000, `PixelFormatRgb16` = 0x02000000, `PixelFormatSignedMono16` = 0x01000000, `PixelFormatSignedRgb16` = 0x00800000, `PixelFormatRaw8` = 0x00400000, `PixelFormatRaw16` = 0x00200000, `PixelFormatMono12` = 0x00100000, `PixelFormatRaw12` = 0x00080000, `PixelFormatBgr` = 0x80000008, `PixelFormatBgru` = 0x40000008, `PixelFormatRgb` = `PixelFormatRgb8`, `PixelFormatRgbu` = 0x40000002, `PixelFormatBgr16` = 0x02000001, `PixelFormatBgru16` = 0x02000002, `PixelFormat422Yuv8Jpeg` = 0x40000001, `NumberOfPixelFormats` = 20 }

Pixel formats available for Format7 modes.

- enum `BusSpeed` { `S100`, `S200`, `S400`, `S480`, `S800`, `S1600`, `S3200`, `S5000`, `GigE_10Base_T`, `GigE_100Base_T`, `GigE_1000Base_T`, `GigE_10000Base_T`, `Fastest`, `Any`, `Unknown` = -1 }

Bus speeds.

- enum `PCleBusSpeed` { `Speed_2_5`, `Speed_5_0`, `Unknown` = -1 }

PCle Bus Speeds.

- enum `DriverType` { `ieee1394_Cam`, `ieee1394_Pro`, `ieee1394_Juju`, `ieee1394_Video1394`, `ieee1394_Raw1394`, `Usb_None`, `Usb_Cam`, `Usb3_Pro`, `GigE_None`, `GigE_Filter`, `GigE_Pro`, `Unknown` = -1 }

Types of low level drivers that flycapture uses.

- enum `ColorProcessingAlgorithm` { `Default`, `NoColorProcessing`, `NearestNeighbor`, `EdgeSensing`, `HQLinear`, `Rigorous`, `IPP`, `Directional` }

Color processing algorithms.

- enum `BayerTileFormat` { `None` = 0, `RGGB`, `GRBG`, `GBRG`, `BGGR` }

Bayer tile formats.

- enum `ImageFileFormat` { `FromFileExtension` = -1, `Pgm`, `Ppm`, `Bmp`, `Jpeg`, `Jpeg2000`, `Tiff`, `Png`, `Raw` }

File formats to be used for saving images to disk.

- enum `StatisticsChannel` { `Grey`, `Red`, `Green`, `Blue`, `Hue`, `Saturation`, `Lightness`, `NumberOfStatisticsChannels` }

Channels that allow statistics to be calculated.

- enum `OSType` { `WindowsX86`, `WindowsX64`, `LinuxX86`, `LinuxX64`, `Mac`, `UnknownOS` }

Possible operating systems.

- enum `ByteOrder` { `LittleEndian`, `BigEndian` }

Possible byte orders.

- enum `GigEPropertyType` { `Heartbeat`, `HeartbeatTimeout`, `PacketSize`, `PacketDelay` }

Possible properties that can be queried from the camera.

5.1.1 Enumeration Type Documentation

5.1.1.1 enum BandwidthAllocation

Bandwidth allocation options for 1394 devices.

Enumerator:

Off Do not allocate bandwidth.

On Allocate bandwidth. This is the default setting.

Unsupported Bandwidth allocation is not supported by either the camera or operating system.

Unspecified Unspecified grab mode. Unspecified property type.

Not specified.

Unspecified timeout setting.

This leaves the current setting unchanged.

5.1.1.2 enum BayerTileFormat

Bayer tile formats.

Enumerator:

None Non-blocking wait. No bayer tile format.

RGGB Red-Green-Green-Blue.

GRBG Green-Red-Blue-Green.

GBRG Green-Blue-Red-Green.

BGGR Blue-Green-Green-Red.

5.1.1.3 enum BusSpeed

Bus speeds.

Enumerator:

- S100** 100Mbps/sec.
- S200** 200Mbps/sec.
- S400** 400Mbps/sec.
- S480** 480Mbps/sec. Only for USB2 cameras.
- S800** 800Mbps/sec.
- S1600** 1600Mbps/sec.
- S3200** 3200Mbps/sec.
- S5000** 5000Mbps/sec. Only for USB3 cameras.
- GigE_10Base_T**
- GigE_100Base_T**
- GigE_1000Base_T**
- GigE_10000Base_T**
- Fastest** The fastest speed available.
- Any** Any speed that is available.
- Unknown** Unknown interface. Unknown driver type.
 - 5.0 Gb/s
 - Unknown bus speed.
 - Speed is unknown

5.1.1.4 enum ByteOrder

Possible byte orders.

Enumerator:

- LittleEndian**
- BigEndian**

5.1.1.5 enum ColorProcessingAlgorithm

Color processing algorithms.

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

Enumerator:

- Default** Default method.
- NoColorProcessing** No color processing.
- NearestNeighbor** Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.
- EdgeSensing** Weights surrounding pixels based on localized edge orientation.
- HQLinear** Similar quality to rigorous but much faster.
- Rigorous** Slowest but produces the best results.
- IPP** Multithreaded with similar results to edge sensing.
- Directional** Best quality but much faster than rigorous.

5.1.1.6 enum DriverType

Types of low level drivers that flycapture uses.

Enumerator:

- lee1394_Cam** PGRCam.sys.
- lee1394_Pro** PGR1394.sys.
- lee1394_Juju** firewire_core.
- lee1394_Video1394** video1394.
- lee1394_Raw1394** raw1394.
- Usb_None** No usb driver used just BSD stack. (Linux only)
- Usb_Cam** PGRUsbCam.sys.
- Usb3_Pro** PGRXHCl.sys.
- GigE_None** no gige drivers used,MS/BSD stack.
- GigE_Filter** PGRGigE.sys.
- GigE_Pro** PGRGigEPro.sys.
- Unknown** Unknown interface. Unknown driver type.
 - 5.0 Gb/s
 - Unknown bus speed.
 - Speed is unknown

5.1.1.7 enum ErrorType

The error types returned by functions.

Enumerator:

- Undefined** Undefined.
- Ok** Function returned with no errors.

Failed General failure.

NotImplemented Function has not been implemented.

FailedBusMasterConnection Could not connect to Bus Master.

NotConnected Camera has not been connected.

InitFailed Initialization failed.

NotInitialized Camera has not been initialized.

InvalidParameter Invalid parameter passed to function.

InvalidSettings Setting set to camera is invalid.

InvalidBuManager Invalid Bus Manager object.

MemoryAllocationFailed Could not allocate memory.

LowLevelFailure Low level error.

NotFound Device not found.

FailedGuid GUID failure.

InvalidPacketSize Packet size set to camera is invalid.

InvalidMode Invalid mode has been passed to function.

NotInFormat7 Error due to not being in Format7.

NotSupported This feature is unsupported.

Timeout Timeout error.

BusMasterFailed Bus Master Failure.

InvalidGeneration Generation Count Mismatch.

LutFailed Look Up Table failure.

IidcFailed IIDC failure.

StrobeFailed Strobe failure.

TriggerFailed Trigger failure.

PropertyFailed Property failure.

PropertyNotPresent Property is not present.

RegisterFailed Register access failed.

ReadRegisterFailed Register read failed.

WriteRegisterFailed Register write failed.

IsochFailed Isochronous failure.

IsochAlreadyStarted Isochronous transfer has already been started.

IsochNotStarted Isochronous transfer has not been started.

IsochStartFailed Isochronous start failed.

IsochRetrieveBufferFailed Isochronous retrieve buffer failed.

IsochStopFailed Isochronous stop failed.

IsochSyncFailed Isochronous image synchronization failed.

IsochBandwidthExceeded Isochronous bandwidth exceeded.

ImageConversionFailed Image conversion failed.

ImageLibraryFailure Image library failure.

BufferTooSmall Buffer is too small.

ImageConsistencyError There is an image consistency error.

5.1.1.8 enum FrameRate

Frame rates in frames per second.

Enumerator:

FrameRate1_875 1.875 fps.
FrameRate3_75 3.75 fps.
FrameRate7_5 7.5 fps.
FrameRate15 15 fps.
FrameRate30 30 fps.
FrameRate60 60 fps.
FrameRate120 120 fps.
FrameRate240 240 fps.
FrameRateFormat7 Custom frame rate for Format7 functionality.
NumberOfFrameRates Number of possible camera frame rates.

5.1.1.9 enum GigEPropertyType

Possible properties that can be queried from the camera.

Enumerator:

Heartbeat
HeartbeatTimeout
PacketSize
PacketDelay

5.1.1.10 enum GrabMode

The grab strategy employed during image transfer.

This type controls how images that stream off the camera accumulate in a user buffer for handling. Unlike earlier versions of the FlyCapture SDK, it is no longer necessary to explicitly start the image grabbing process before specifying an image grabbing mode.

Enumerator:

DropFrames Grabs the newest image in the user buffer each time the Retrieve-Buffer() function is called. Older images are dropped instead of accumulating in the user buffer. Grabbing blocks if the camera has not finished transmitting the next available image. If the camera is transmitting images faster than the application can grab them, images may be dropped and only the most recent image is stored for grabbing. Note that this mode is the equivalent of flycaptureLockLatest in earlier versions of the FlyCapture SDK.

BufferFrames Images accumulate in the user buffer, and the oldest image is grabbed for handling before being discarded. This member can be used to guarantee that each image is seen. However, image processing time must not exceed transmission time from the camera to the buffer. Grabbing blocks if the camera has not finished transmitting the next available image. The buffer size is controlled by the numBuffers parameter in the [FC2Config](#) struct. Note that this mode is the equivalent of flycaptureLockNext in earlier versions of the FlyCapture SDK.

Unspecified Unspecified grab mode. Unspecified property type.

Not specified.

Unspecified timeout setting.

This leaves the current setting unchanged.

5.1.1.11 enum GrabTimeout

Timeout options for grabbing images.

Enumerator:

None Non-blocking wait. No bayer tile format.

Infinite Wait indefinitely.

Unspecified Unspecified grab mode. Unspecified property type.

Not specified.

Unspecified timeout setting.

This leaves the current setting unchanged.

5.1.1.12 enum ImageFileFormat

File formats to be used for saving images to disk.

Enumerator:

FromFileExtension Determine file format from file extension.

Pgm Portable gray map.

Ppm Portable pixmap.

Bmp Bitmap.

Jpeg JPEG.

Jpeg2000 JPEG 2000.

Tiff Tagged image file format.

Png Portable network graphics.

Raw Raw data.

5.1.1.13 enum InterfaceType

Interfaces that a camera may use to communicate with a host.

Enumerator:

- IEEE1394** IEEE-1394 (Includes 1394a and 1394b).
- Usb2** USB 2.0.
- Usb3** USB 3.0.
- GigE** GigE.
- Unknown** Unknown interface. Unknown driver type.
 - 5.0 Gb/s
 - Unknown bus speed.
 - Speed is unknown

5.1.1.14 enum ManagedCallbackType

The type of bus callback to register a callback function for.

Enumerator:

- BusReset** Register for all bus events.
- Arrival** Register for arrivals only.
- Removal** Register for removals only.

5.1.1.15 enum Mode

Camera modes for DCAM formats as well as Format7.

Enumerator:

- Mode0**
- Mode1**
- Mode2**
- Mode3**
- Mode4**
- Mode5**
- Mode6**
- Mode7**
- Mode8**
- Mode9**
- Mode10**

Mode11
Mode12
Mode13
Mode14
Mode15
Mode16
Mode17
Mode18
Mode19
Mode20
Mode21
Mode22
Mode23
Mode24
Mode25
Mode26
Mode27
Mode28
Mode29
Mode30
Mode31
NumberOfModes

5.1.1.16 enum OSType

Possible operating systems.

Enumerator:

WindowsX86 All Windows 32-bit variants.
WindowsX64 All Windows 64-bit variants.
LinuxX86 All Linux 32-bit variants.
LinuxX64 All Linux 32-bit variants.
Mac Mac OSX.
UnknownOS Unknown operating system.

5.1.1.17 enum PCIeBusSpeed

PCIe Bus Speeds.

Enumerator:

Speed_2_5

Speed_5_0 2.5 Gb/s

Unknown Unknown interface. Unknown driver type.

5.0 Gb/s

Unknown bus speed.

Speed is unknown

5.1.1.18 enum PixelFormat

Pixel formats available for Format7 modes.

Enumerator:

PixelFormatMono8 8 bits of mono information.

PixelFormat411Yuv8 YUV 4:1:1.

PixelFormat422Yuv8 YUV 4:2:2.

PixelFormat444Yuv8 YUV 4:4:4.

PixelFormatRgb8 R = G = B = 8 bits.

PixelFormatMono16 16 bits of mono information.

PixelFormatRgb16 R = G = B = 16 bits.

PixelFormatSignedMono16 16 bits of signed mono information.

PixelFormatSignedRgb16 R = G = B = 16 bits signed.

PixelFormatRaw8 8 bit raw data output of sensor.

PixelFormatRaw16 16 bit raw data output of sensor.

PixelFormatMono12 12 bits of mono information.

PixelFormatRaw12 12 bit raw data output of sensor.

PixelFormatBgr 24 bit BGR.

PixelFormatBgru 32 bit BGRU.

PixelFormatRgb 24 bit RGB.

PixelFormatRgbu 32 bit RGBU.

PixelFormatBgr16 R = G = B = 16 bits.

PixelFormatBgru16 64 bit BGRU.

PixelFormat422Yuv8Jpeg JPEG compressed stream.

NumberOfPixelFormat Number of pixel formats.

5.1.1.19 enum PropertyType

Camera properties.

Not all properties may be supported, depending on the camera model.

Enumerator:

Brightness Brightness.
AutoExposure Auto exposure.
Sharpness Sharpness.
WhiteBalance White balance.
Hue Hue.
Saturation Saturation.
Gamma Gamma.
Iris Iris.
Focus Focus.
Zoom Zoom.
Pan Pan.
Tilt Tilt.
Shutter Shutter.
Gain Gain.
TriggerMode Trigger mode.
TriggerDelay Trigger delay.
FrameRate Frame rate.
Temperature Temperature.
Unspecified Unspecified grab mode. Unspecified property type.
Not specified.
Unspecified timeout setting.
This leaves the current setting unchanged.

5.1.1.20 enum StatisticsChannel

Channels that allow statistics to be calculated.

Enumerator:

Grey
Red
Green
Blue
Hue Hue.
Saturation Saturation.
Lightness
NumberOfStatisticsChannels

5.1.1.21 enum VideoMode

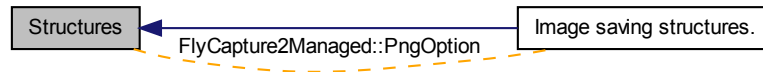
DCAM video modes.

Enumerator:

VideoMode160x120Yuv444 160x120 YUV444.
VideoMode320x240Yuv422 320x240 YUV422.
VideoMode640x480Yuv411 640x480 YUV411.
VideoMode640x480Yuv422 640x480 YUV422.
VideoMode640x480Rgb 640x480 24-bit RGB.
VideoMode640x480Y8 640x480 8-bit.
VideoMode640x480Y16 640x480 16-bit.
VideoMode800x600Yuv422 800x600 YUV422.
VideoMode800x600Rgb 800x600 RGB.
VideoMode800x600Y8 800x600 8-bit.
VideoMode800x600Y16 800x600 16-bit.
VideoMode1024x768Yuv422 1024x768 YUV422.
VideoMode1024x768Rgb 1024x768 RGB.
VideoMode1024x768Y8 1024x768 8-bit.
VideoMode1024x768Y16 1024x768 16-bit.
VideoMode1280x960Yuv422 1280x960 YUV422.
VideoMode1280x960Rgb 1280x960 RGB.
VideoMode1280x960Y8 1280x960 8-bit.
VideoMode1280x960Y16 1280x960 16-bit.
VideoMode1600x1200Yuv422 1600x1200 YUV422.
VideoMode1600x1200Rgb 1600x1200 RGB.
VideoMode1600x1200Y8 1600x1200 8-bit.
VideoMode1600x1200Y16 1600x1200 16-bit.
VideoModeFormat7 Custom video mode for Format7 functionality.
NumberOfVideoModes Number of possible video modes.

5.2 Structures

Collaboration diagram for Structures:



Classes

- struct [FC2Version](#)
The current version of the library.
- struct [GigEProperty](#)
A GigE property.
- struct [GigEStreamChannel](#)
Information about a single GigE stream channel.
- struct [GigEConfig](#)
Configuration for a GigE camera.
- struct [GigEImageSettingsInfo](#)
Format 7 information for a single mode.
- struct [GigEImageSettings](#)
Image settings for a GigE camera.
- struct [FC2Config](#)
Configuration for a camera.
- struct [CameraPropertyInfo](#)
Information about a specific camera property.
- struct [CameraProperty](#)
A specific camera property.
- struct [TriggerModelInfo](#)
Information about a camera trigger property.
- struct [TriggerMode](#)
A camera trigger.
- struct [StrobeInfo](#)
A camera strobe property.
- struct [StrobeControl](#)
A camera strobe.
- struct [Format7ImageSettings](#)
Format 7 image settings.
- struct [Format7Info](#)

- *Format 7 information for a single mode.*
• struct [Format7PacketInfo](#)
Format 7 packet information.
- struct [TimeStamp](#)
Timestamp information.
- struct [ConfigROM](#)
Camera configuration ROM.
- struct [CameraInfo](#)
Camera information.
- struct [EmbeddedImageInfoProperty](#)
Properties of a single embedded image info property.
- struct [EmbeddedImageInfo](#)
Properties of the possible embedded image information.
- struct [ImageMetadata](#)
Metadata related to an image.
- struct [LutData](#)
Information about the camera's look up table.
- struct [PngOption](#)
Options for saving PNG images.

Modules

- [Image saving structures.](#)
These structures define various parameters used for saving images.

5.3 Image saving structures.

These structures define various parameters used for saving images.

Collaboration diagram for Image saving structures.:



Classes

- struct [PngOption](#)
Options for saving PNG images.
- struct [PpmOption](#)
Options for saving PPM images.
- struct [PgmOption](#)
Options for saving PGM images.
- struct [TiffOption](#)
Options for saving TIFF images.
- struct [JpegOption](#)
Options for saving JPEG image.
- struct [Jpg2Option](#)
Options for saving JPEG2000 image.
- struct [AviOption](#)
Options for saving AVI files.
- struct [MJPGOption](#)
Options for saving MJPEG files.
- struct [H264Option](#)
Options for saving H.264 files.
- struct [SystemInfo](#)
Description of the system.

5.3.1 Detailed Description

These structures define various parameters used for saving images.

Chapter 6

Namespace Documentation

6.1 FlyCapture2 Namespace Reference

6.2 FlyCapture2Managed Namespace Reference

Namespaces

- namespace [Gui](#)

Classes

- class [FC2Exception](#)
Exception that is thrown when an error is encountered.
- class [ManagedAVIRecorder](#)
[ManagedAVIRecorder](#) provides the functionality for the user to record images to an AVI file.
- class [ManagedBusManager](#)
[ManagedBusManager](#) provides the functionality for the user to get an PGRGuid for a desired camera or device easily.
- class [ManagedCamera](#)
[ManagedCamera](#) represents a physical camera that uses the IIDC register set.
- class [ManagedCameraBase](#)
Abstract base class that represents a generic camera that defines a general interface to a camera.
- struct [FC2Version](#)
The current version of the library.
- struct [GigEProperty](#)
A GigE property.
- struct [GigEStreamChannel](#)
Information about a single GigE stream channel.

- struct [GigEConfig](#)
Configuration for a GigE camera.
- struct [GigEImageSettingsInfo](#)
Format 7 information for a single mode.
- struct [GigEImageSettings](#)
Image settings for a GigE camera.
- struct [FC2Config](#)
Configuration for a camera.
- struct [CameraPropertyInfo](#)
Information about a specific camera property.
- struct [CameraProperty](#)
A specific camera property.
- struct [TriggerModelInfo](#)
Information about a camera trigger property.
- struct [TriggerMode](#)
A camera trigger.
- struct [StrobeInfo](#)
A camera strobe property.
- struct [StrobeControl](#)
A camera strobe.
- struct [Format7ImageSettings](#)
Format 7 image settings.
- struct [Format7Info](#)
Format 7 information for a single mode.
- struct [Format7PacketInfo](#)
Format 7 packet information.
- struct [TimeStamp](#)
Timestamp information.
- struct [ConfigROM](#)
Camera configuration ROM.
- struct [CameraInfo](#)
Camera information.
- struct [EmbeddedImageInfoProperty](#)
Properties of a single embedded image info property.
- struct [EmbeddedImageInfo](#)
Properties of the possible embedded image information.
- struct [ImageMetadata](#)
Metadata related to an image.
- struct [LutData](#)
Information about the camera's look up table.
- struct [PngOption](#)
Options for saving PNG images.
- struct [PpmOption](#)

- Options for saving PPM images.
- struct [PgmOption](#)
- Options for saving PGM images.
- struct [TiffOption](#)
- Options for saving TIFF images.
- struct [JpegOption](#)
- Options for saving JPEG image.
- struct [Jpg2Option](#)
- Options for saving JPEG2000 image.
- struct [AviOption](#)
- Options for saving AVI files.
- struct [MJPGOption](#)
- Options for saving MJPEG files.
- struct [H264Option](#)
- Options for saving H.264 files.
- struct [SystemInfo](#)
- Description of the system.
- class [ManagedGCCamera](#)
- class [ManagedGCPort](#)
- class [ManagedGigECamera](#)
- The [GigECamera](#) object represents a physical Gigabit Ethernet camera.
- class [ManagedImage](#)
- The [ManagedImage](#) class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.
- class [ManagedImageStatistics](#)
- class [ManagedPGRGuid](#)
- Managed version of a [PGRGuid](#).
- class [ManagedTopologyNode](#)
- The [ManagedTopologyNode](#) class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.
- class [ManagedUtilities](#)
- class [Translate](#)

Enumerations

- enum [ErrorType](#) { [Undefined](#) = -1, [Ok](#), [Failed](#), [NotImplemented](#), [FailedBusMasterConnection](#), [NotConnected](#), [InitFailed](#), [NotInitialized](#), [InvalidParameter](#), [InvalidSettings](#), [InvalidBuManager](#), [MemoryAllocationFailed](#), [LowLevelFailure](#), [NotFound](#), [FailedGuid](#), [InvalidPacketSize](#), [InvalidMode](#), [NotInFormat7](#), × [NotSupported](#), [Timeout](#), [BusMasterFailed](#), [InvalidGeneration](#), [LutFailed](#), × [IldcFailed](#), [StrobeFailed](#), [TriggerFailed](#), [PropertyFailed](#), [PropertyNotPresent](#), [RegisterFailed](#), [ReadRegisterFailed](#), [WriteRegisterFailed](#), [IsochFailed](#), × [IsochAlreadyStarted](#), [IsochNotStarted](#), [IsochStartFailed](#), [IsochRetrieveBufferFailed](#), [IsochStopFailed](#), [IsochSyncFailed](#), [IsochBandwidthExceeded](#), [ImageConversionFailed](#), [ImageLibraryFailure](#), [BufferTooSmall](#), [ImageConsistencyError](#) }

The error types returned by functions.

- enum [ManagedCallbackType](#) { [BusReset](#), [Arrival](#), [Removal](#) }

The type of bus callback to register a callback function for.

- enum [GrabMode](#) { [DropFrames](#), [BufferFrames](#), [Unspecified](#) = -2 }

The grab strategy employed during image transfer.

- enum [GrabTimeout](#) { [None](#) = 0, [Infinite](#) = -1, [Unspecified](#) = -2 }

Timeout options for grabbing images.

- enum [BandwidthAllocation](#) { [Off](#) = 0, [On](#) = 1, [Unsupported](#) = 2, [Unspecified](#) = -2 }

Bandwidth allocation options for 1394 devices.

- enum [InterfaceType](#) { [Ieee1394](#), [Usb2](#), [Usb3](#), [GigE](#), [Unknown](#) = -1 }

Interfaces that a camera may use to communicate with a host.

- enum [PropertyType](#) { [Brightness](#), [AutoExposure](#), [Sharpness](#), [WhiteBalance](#), [Hue](#), [Saturation](#), [Gamma](#), [Iris](#), [Focus](#), [Zoom](#), [Pan](#), [Tilt](#), [Shutter](#), [Gain](#), [TriggerMode](#), [TriggerDelay](#), [FrameRate](#), [Temperature](#), [Unspecified](#) = -2 }

Camera properties.

- enum [FrameRate](#) { [FrameRate1_875](#), [FrameRate3_75](#), [FrameRate7_5](#), [FrameRate15](#), [FrameRate30](#), [FrameRate60](#), [FrameRate120](#), [FrameRate240](#), [FrameRateFormat7](#), [NumberOfFrameRates](#) }

Frame rates in frames per second.

- enum [VideoMode](#) { [VideoMode160x120Yuv444](#), [VideoMode320x240Yuv422](#), [VideoMode640x480Yuv411](#), [VideoMode640x480Yuv422](#), [VideoMode640x480Rgb](#), [VideoMode640x480Y8](#), [VideoMode640x480Y16](#), [VideoMode800x600Yuv422](#), [VideoMode800x600Rgb](#), [VideoMode800x600Y8](#), [VideoMode800x600Y16](#), [VideoMode1024x768Yuv422](#), [VideoMode1024x768Rgb](#), [VideoMode1024x768Y8](#), [VideoMode1024x768Y16](#), [VideoMode1280x960Yuv422](#), [VideoMode1280x960Rgb](#), [VideoMode1280x960Y8](#), [VideoMode1280x960Y16](#), [VideoMode1600x1200Yuv422](#), [VideoMode1600x1200Rgb](#), [VideoMode1600x1200Y8](#), [VideoMode1600x1200Y16](#), [VideoModeFormat7](#), [NumberOfVideoModes](#) }

DCAM video modes.

- enum [Mode](#) { [Mode0](#) = 0, [Mode1](#), [Mode2](#), [Mode3](#), [Mode4](#), [Mode5](#), [Mode6](#), [Mode7](#), [Mode8](#), [Mode9](#), [Mode10](#), [Mode11](#), [Mode12](#), [Mode13](#), [Mode14](#), [Mode15](#), [Mode16](#), [Mode17](#), [Mode18](#), [Mode19](#), [Mode20](#), [Mode21](#), [Mode22](#), [Mode23](#), [Mode24](#), [Mode25](#), [Mode26](#), [Mode27](#), [Mode28](#), [Mode29](#), [Mode30](#), [Mode31](#), [NumberOfModes](#) }

Camera modes for DCAM formats as well as Format7.

- enum [PixelFormat](#) { [PixelFormatMono8](#) = 0x80000000, [PixelFormat411Yuv8](#) = 0x40000000, [PixelFormat422Yuv8](#) = 0x20000000, [PixelFormat444Yuv8](#) = 0x10000000, [PixelFormatRgb8](#) = 0x08000000, [PixelFormatMono16](#) = 0x04000000, [PixelFormatRgb16](#) = 0x02000000, [PixelFormatSignedMono16](#) = 0x01000000, [PixelFormatSignedRgb16](#) = 0x00800000, [PixelFormatRaw8](#) = 0x00400000, [PixelFormatRaw16](#) = 0x00200000, [PixelFormatMono12](#) = 0x00100000, [PixelFormatRaw12](#) = 0x00080000, [PixelFormatBgr](#) = 0x80000008, [PixelFormatBgru](#) = 0x40000008, [PixelFormatRgb](#) = [PixelFormatRgb8](#), [PixelFormatRgbu](#) = 0x40000002, [PixelFormatBgr16](#) = 0x02000001, [PixelFormatBgru16](#) = 0x02000002, [PixelFormat422Yuv8Jpeg](#) = 0x40000001, [NumberOfPixelFormats](#) = 20 }

Pixel formats available for Format7 modes.

- enum [BusSpeed](#) { [S100](#), [S200](#), [S400](#), [S480](#), [S800](#), [S1600](#), [S3200](#), [S5000](#), [GigE_10Base_T](#), [GigE_100Base_T](#), [GigE_1000Base_T](#), [GigE_10000Base_T](#), [Fastest](#), [Any](#), [Unknown](#) = -1 }

Bus speeds.

- enum [PCleBusSpeed](#) { [Speed_2_5](#), [Speed_5_0](#), [Unknown](#) = -1 }

PCle Bus Speeds.

- enum [DriverType](#) { [Ieee1394_Cam](#), [Ieee1394_Pro](#), [Ieee1394_Juju](#), [Ieee1394_Video1394](#), [Ieee1394_Raw1394](#), [Usb_None](#), [Usb_Cam](#), [Usb3_Pro](#), [GigE_None](#), [GigE_Filter](#), [GigE_Pro](#), [Unknown](#) = -1 }

Types of low level drivers that flycapture uses.

- enum [ColorProcessingAlgorithm](#) { [Default](#), [NoColorProcessing](#), [NearestNeighbor](#), [EdgeSensing](#), [HQLinear](#), [Rigorous](#), [IPP](#), [Directional](#) }

Color processing algorithms.

- enum [BayerTileFormat](#) { [None](#) = 0, [RGGB](#), [GRBG](#), [GBRG](#), [BGGR](#) }

Bayer tile formats.

- enum [ImageFileFormat](#) { [FromFileExtension](#) = -1, [Pgm](#), [Ppm](#), [Bmp](#), [Jpeg](#), [Jpeg2000](#), [Tiff](#), [Png](#), [Raw](#) }

File formats to be used for saving images to disk.

- enum [StatisticsChannel](#) { [Grey](#), [Red](#), [Green](#), [Blue](#), [Hue](#), [Saturation](#), [Lightness](#), [NumberOfStatisticsChannels](#) }

Channels that allow statistics to be calculated.

- enum [OSType](#) { [WindowsX86](#), [WindowsX64](#), [LinuxX86](#), [LinuxX64](#), [Mac](#), [UnknownOS](#) }

Possible operating systems.

- enum [ByteOrder](#) { [LittleEndian](#), [BigEndian](#) }

Possible byte orders.

- enum [GigEPropertyType](#) { [Heartbeat](#), [HeartbeatTimeout](#), [PacketSize](#), [PacketDelay](#) }

Possible properties that can be queried from the camera.

Functions

- public delegate void [EnumCallback](#) (System::IntPtr parameter, unsigned int serialNumber)

Bus event callback function prototype.

- public delegate void [ImageEventCallback](#) ([ManagedImage](#)[^] image)

The external callback that will be used by managed consumers.

- protected delegate void [ImageCallbackDelegate](#) (FlyCapture2::Image *image, void *data)

Internal callback that we use internally so we can create the proper proper external callback for users.

6.2.1 Function Documentation

6.2.1.1 `public delegate void FlyCapture2Managed::EnumCallback (System::IntPtr parameter,
unsigned int serialNumber)`

Bus event callback function prototype.

Defines the syntax of the callback function that is passed into RegisterCallback() and UnregisterCallback().

6.2.1.2 `protected delegate void FlyCapture2Managed::ImageCallbackDelegate (FlyCapture2::Image * image, void * data)`

Internal callback that we use internally so we can create the proper external callback for users.

6.2.1.3 `public delegate void FlyCapture2Managed::ImageEventCallback (ManagedImage^ image)`

The external callback that will be used by managed consumers.

6.3 FlyCapture2Managed::Gui Namespace Reference

Classes

- class [CameraControlDialog](#)
CameraControlDialog: managed wrapper of FlyCapture2::CameraControlDialog (see for details)
- class [CameraSelectionDialog](#)
CameraControlDialog: managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)

Chapter 7

Class Documentation

7.1 AviOption Struct Reference

Options for saving AVI files.

Public Member Functions

- [AviOption](#) ()

Properties

- float [frameRate](#)

Frame rate of the stream.

7.1.1 Detailed Description

Options for saving AVI files.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 [AviOption](#) () `[inline]`

7.1.3 Property Documentation

7.1.3.1 float [frameRate](#)

Frame rate of the stream.

7.2 CameraControlDialog Class Reference

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraControlDialog (see for details)

Public Member Functions

- [CameraControlDialog](#) ()
- [~CameraControlDialog](#) ()
- void [Connect](#) ([ManagedCameraBase](#)^ camera)
- void [Disconnect](#) ()
- void [Show](#) ()
- void [Hide](#) ()
- bool [IsVisible](#) ()
- void [SetTitle](#) (System::String^ title)

7.2.1 Detailed Description

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraControlDialog (see for details)

7.2.2 Constructor & Destructor Documentation

7.2.2.1 [CameraControlDialog](#) ()

7.2.2.2 [~CameraControlDialog](#) ()

7.2.3 Member Function Documentation

7.2.3.1 void [Connect](#) ([FlyCapture2Managed::ManagedCameraBase](#)^ camera)

7.2.3.2 void [Disconnect](#) (void)

7.2.3.3 void [Hide](#) ()

7.2.3.4 bool [IsVisible](#) ()

7.2.3.5 void [SetTitle](#) (System::String^ title)

7.2.3.6 void [Show](#) ()

7.3 CameraInfo Struct Reference

Camera information.

Properties

- unsigned int [serialNumber](#)
Device serial number.
- [InterfaceType](#) [interfaceType](#)
Interface type.
- [DriverType](#) [driverType](#)
Driver type.
- bool [isColorCamera](#)
Flag indicating if this is a color camera.
- System::String^ [modelName](#)
Device model name.
- System::String^ [vendorName](#)
Device vendor name.
- System::String^ [sensorInfo](#)
String detailing the sensor information.
- System::String^ [sensorResolution](#)
String providing the sensor resolution.
- System::String^ [driverName](#)
Driver name of driver being used.
- System::String^ [firmwareVersion](#)
Firmware version of camera.
- System::String^ [firmwareBuildTime](#)
Firmware build time.
- [BusSpeed](#) [maximumBusSpeed](#)
Maximum bus speed.
- [PCleBusSpeed](#) [pcieBusSpeed](#)
Maximum PCIe bus speed.
- [BayerTileFormat](#) [bayerTileFormat](#)
Bayer tile format.
- unsigned short [busNumber](#)
Bus Number, set to 0 for USB and GigE.
- unsigned short [nodeNumber](#)
Node Number, set to 0 for USB and GigE.

IIDC specific information

- unsigned int [iidcVersion](#)
DCAM version.
- [ConfigROM](#) [configROM](#)
Configuration ROM data.

GigE specific information

- unsigned int [gigEMajorVersion](#)
GigE Vision version.
- unsigned int [gigEMinorVersion](#)
GigE Vision minor version.
- System::String^ [userDefinedName](#)
User defined name.
- System::String^ [xmlURL1](#)
XML URL 1.
- System::String^ [xmlURL2](#)
XML URL 2.
- System::Net::NetworkInformation::PhysicalAddress^ [macAddress](#)
MAC address.
- System::Net::IPAddress^ [ipAddress](#)
IP address.
- System::Net::IPAddress^ [subnetMask](#)
Subnet mask.
- System::Net::IPAddress^ [defaultGateway](#)
Default gateway.
- unsigned int [ccpStatus](#)
Status/Content of CCP register.
- unsigned int [applicationIPAddress](#)
Local Application IP Address.
- unsigned int [applicationPort](#)
Local Application port.

7.3.1 Detailed Description

Camera information.

7.3.2 Property Documentation**7.3.2.1 unsigned int applicationIPAddress**

Local Application IP Address.

7.3.2.2 unsigned int applicationPort

Local Application port.

7.3.2.3 BayerTileFormat bayerTileFormat

Bayer tile format.

7.3.2.4 unsigned short busNumber

Bus Number, set to 0 for USB and GigE.

7.3.2.5 unsigned int ccpStatus

Status/Content of CCP register.

7.3.2.6 ConfigROM configROM

Configuration ROM data.

7.3.2.7 System:: Net:: IPAddress^ defaultGateway

Default gateway.

7.3.2.8 System:: String^ driverName

Driver name of driver being used.

7.3.2.9 DriverType driverType

Driver type.

7.3.2.10 System:: String^ firmwareBuildTime

Firmware build time.

7.3.2.11 System:: String^ firmwareVersion

Firmware version of camera.

7.3.2.12 unsigned int gigEMajorVersion

GigE Vision version.

7.3.2.13 unsigned int gigEMinorVersion

GigE Vision minor version.

7.3.2.14 unsigned int iidcVersion

DCAM version.

7.3.2.15 InterfaceType interfaceType

Interface type.

7.3.2.16 System:: Net:: IPAddress^ ipAddress

IP address.

7.3.2.17 bool isColorCamera

Flag indicating if this is a color camera.

7.3.2.18 System:: Net:: NetworkInformation:: PhysicalAddress^ macAddress

MAC address.

7.3.2.19 BusSpeed maximumBusSpeed

Maximum bus speed.

7.3.2.20 System:: String^ modelName

Device model name.

7.3.2.21 unsigned short nodeNumber

Node Number, set to 0 for USB and GigE.

7.3.2.22 PCIeBusSpeed pcieBusSpeed

Maximum PCIe bus speed.

7.3.2.23 System:: String^ sensorInfo

String detailing the sensor information.

7.3.2.24 System:: String^ sensorResolution

String providing the sensor resolution.

7.3.2.25 unsigned int serialNumber

Device serial number.

7.3.2.26 System:: Net:: IPAddress^ subnetMask

Subnet mask.

7.3.2.27 System:: String^ userDefinedName

User defined name.

7.3.2.28 System:: String^ vendorName

Device vendor name.

7.3.2.29 System:: String^ xmlURL1

XML URL 1.

7.3.2.30 System:: String^ xmlURL2

XML URL 2.

7.4 CameraProperty Struct Reference

A specific camera property.

Public Member Functions

- [CameraProperty](#) ()
- [CameraProperty](#) (PropertyType type)

Properties

- [PropertyType](#) type
Property info type.

- bool `present`
Flag indicating if the property is present.
- bool `absControl`
Flag controlling absolute mode.
- bool `onePush`
Flag controlling one push.
- bool `onOff`
Flag controlling on/off.
- bool `autoManualMode`
Flag controlling auto.
- unsigned int `valueA`
Value A (integer).
- unsigned int `valueB`
Value B (integer).
- float `absValue`
Floating point value.

7.4.1 Detailed Description

A specific camera property.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 `CameraProperty ()` `[inline]`

7.4.2.2 `CameraProperty (PropertyType type)` `[inline]`

7.4.3 Property Documentation

7.4.3.1 `bool absControl`

Flag controlling absolute mode.

7.4.3.2 `float absValue`

Floating point value.

7.4.3.3 `bool autoManualMode`

Flag controlling auto.

7.4.3.4 bool onePush

Flag controlling one push.

7.4.3.5 bool onOff

Flag controlling on/off.

7.4.3.6 bool present

Flag indicating if the property is present.

7.4.3.7 PropertyType type

Property info type.

7.4.3.8 unsigned int valueA

Value A (integer).

7.4.3.9 unsigned int valueB

Value B (integer).

Applies only to the white balance blue value. Use Value A for the red value.

7.5 CameraPropertyInfo Struct Reference

Information about a specific camera property.

Public Member Functions

- [CameraPropertyInfo \(\)](#)
- [CameraPropertyInfo \(PropertyType type\)](#)

Properties

- [PropertyType type](#)
Property info type.
- bool [present](#)
Flag indicating if the property is present.
- bool [autoSupported](#)

- Flag indicating if auto is supported.*
 - bool [manualSupported](#)
- Flag indicating if manual is supported.*
 - bool [onOffSupported](#)
- Flag indicating if on/off is supported.*
 - bool [onePushSupported](#)
- Flag indicating if one push is supported.*
 - bool [absValSupported](#)
- Flag indicating if absolute mode is supported.*
 - bool [readOutSupported](#)
- Flag indicating if property value can be read out.*
 - unsigned int [min](#)
- Minimum value (as an integer).*
 - unsigned int [max](#)
- Maximum value (as an integer).*
 - float [absMin](#)
- Minimum value (as a floating point value).*
 - float [absMax](#)
- Maximum value (as a floating point value).*
 - System::String^ [units](#)
- Textual description of units.*
 - System::String^ [unitAbbr](#)
- Abbreviated textual description of units.*

7.5.1 Detailed Description

Information about a specific camera property.

This structure is also also used as the TriggerDelayInfo structure.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 **CameraPropertyInfo ()** *[inline]*

7.5.2.2 **CameraPropertyInfo (PropertyType type)** *[inline]*

7.5.3 Property Documentation

7.5.3.1 **float absMax**

Maximum value (as a floating point value).

7.5.3.2 **float absMin**

Minimum value (as a floating point value).

7.5.3.3 bool absValSupported

Flag indicating if absolute mode is supported.

7.5.3.4 bool autoSupported

Flag indicating if auto is supported.

7.5.3.5 bool manualSupported

Flag indicating if manual is supported.

7.5.3.6 unsigned int max

Maximum value (as an integer).

7.5.3.7 unsigned int min

Minimum value (as an integer).

7.5.3.8 bool onePushSupported

Flag indicating if one push is supported.

7.5.3.9 bool onOffSupported

Flag indicating if on/off is supported.

7.5.3.10 bool present

Flag indicating if the property is present.

7.5.3.11 bool readOutSupported

Flag indicating if property value can be read out.

7.5.3.12 PropertyType type

Property info type.

7.5.3.13 System::String^ unitAbbr

Abbreviated textual description of units.

7.5.3.14 System::String^ units

Textual description of units.

7.6 CameraSelectionDialog Class Reference

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)

Public Member Functions

- [CameraSelectionDialog](#) ()
- [~CameraSelectionDialog](#) ()
- bool [ShowModal](#) ()
Show the CameraSelectionDlg.
- array< [ManagedPGRGuid](#)^ >^ [GetSelectedCameraGuids](#) ()
Returns the list of camera guides selected by the user while in [ShowModal\(\)](#)
- void [SetTitle](#) (System::String^ title)
Set the window title.

7.6.1 Detailed Description

[CameraControlDialog](#): managed wrapper of FlyCapture2::CameraSelectionDialog (see for details)

7.6.2 Constructor & Destructor Documentation

7.6.2.1 CameraSelectionDialog ()

7.6.2.2 ~CameraSelectionDialog ()

7.6.3 Member Function Documentation

7.6.3.1 array< ManagedPGRGuid^ > GetSelectedCameraGuids ()

Returns the list of camera guides selected by the user while in [ShowModal\(\)](#)

Returns

Array of PGRGuids identifying the selected cameras.

7.6.3.2 void SetTitle (System::String^ title)

Set the window title.

Parameters

<i>pTitle</i>	Null-terminated string representing the title.
---------------	--

7.6.3.3 bool ShowModal ()

Show the CameraSelectionDlg.

Returns

Whether Ok (true) or Cancel (false) was clicked.

7.7 ConfigROM Struct Reference

Camera configuration ROM.

Properties

- unsigned int [nodeVendorId](#)
Vendor ID of a node.
- unsigned int [chipIdHi](#)
Chip ID (high part).
- unsigned int [chipIdLo](#)
Chip ID (low part).
- unsigned int [unitSpecId](#)
Unit Spec ID, usually 0xa02d.
- unsigned int [unitSWVer](#)
Unit software version.
- unsigned int [unitSubSWVer](#)
Unit sub software version.
- unsigned int [vendorUniqueInfo0](#)
Vendor unique info 0.
- unsigned int [vendorUniqueInfo1](#)
Vendor unique info 1.
- unsigned int [vendorUniqueInfo2](#)
Vendor unique info 2.
- unsigned int [vendorUniqueInfo3](#)
Vendor unique info 3.
- System::String^ [keyword](#)
Keyword.

7.7.1 Detailed Description

Camera configuration ROM.

7.7.2 Property Documentation

7.7.2.1 unsigned int chipIdHi

Chip ID (high part).

7.7.2.2 unsigned int chipIdLo

Chip ID (low part).

7.7.2.3 System::String^ keyword

Keyword.

7.7.2.4 unsigned int nodeVendorId

Vendor ID of a node.

7.7.2.5 unsigned int unitSpecId

Unit Spec ID, usually 0xa02d.

7.7.2.6 unsigned int unitSubSWVer

Unit sub software version.

7.7.2.7 unsigned int unitSWVer

Unit software version.

7.7.2.8 unsigned int vendorUniqueInfo0

Vendor unique info 0.

7.7.2.9 unsigned int vendorUniqueInfo1

Vendor unique info 1.

7.7.2.10 unsigned int vendorUniqueInfo2

Vendor unique info 2.

7.7.2.11 unsigned int vendorUniqueInfo3

Vendor unique info 3.

7.8 EmbeddedImageInfo Struct Reference

Properties of the possible embedded image information.

Public Member Functions

- [EmbeddedImageInfo \(\)](#)

Properties

- [EmbeddedImageInfoProperty^ timestamp](#)
- [EmbeddedImageInfoProperty^ gain](#)
- [EmbeddedImageInfoProperty^ shutter](#)
- [EmbeddedImageInfoProperty^ brightness](#)
- [EmbeddedImageInfoProperty^ exposure](#)
- [EmbeddedImageInfoProperty^ whiteBalance](#)
- [EmbeddedImageInfoProperty^ frameCounter](#)
- [EmbeddedImageInfoProperty^ strobePattern](#)
- [EmbeddedImageInfoProperty^ GPIOPinState](#)
- [EmbeddedImageInfoProperty^ ROIPosition](#)

7.8.1 Detailed Description

Properties of the possible embedded image information.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 [EmbeddedImageInfo \(\)](#) `[inline]`

7.8.3 Property Documentation

7.8.3.1 [EmbeddedImageInfoProperty^ brightness](#)7.8.3.2 [EmbeddedImageInfoProperty^ exposure](#)

7.8.3.3 `EmbeddedImageInfoProperty`^ `frameCounter`

7.8.3.4 `EmbeddedImageInfoProperty`^ `gain`

7.8.3.5 `EmbeddedImageInfoProperty`^ `GPIOPinState`

7.8.3.6 `EmbeddedImageInfoProperty`^ `ROIPosition`

7.8.3.7 `EmbeddedImageInfoProperty`^ `shutter`

7.8.3.8 `EmbeddedImageInfoProperty`^ `strobePattern`

7.8.3.9 `EmbeddedImageInfoProperty`^ `timestamp`

7.8.3.10 `EmbeddedImageInfoProperty`^ `whiteBalance`

7.9 `EmbeddedImageInfoProperty` Struct Reference

Properties of a single embedded image info property.

Properties

- bool `available`
Whether this property is available.
- bool `onOff`
Whether this property is on or off.

7.9.1 Detailed Description

Properties of a single embedded image info property.

7.9.2 Property Documentation

7.9.2.1 bool `available`

Whether this property is available.

7.9.2.2 bool `onOff`

Whether this property is on or off.

7.10 FC2Config Struct Reference

Configuration for a camera.

Public Member Functions

- [FC2Config](#) ()

Properties

- unsigned int [numBuffers](#)
Number of buffers used by the [FlyCapture2](#) library to grab images.
- unsigned int [numImageNotifications](#)
Number of notifications per image.
- unsigned int [minNumImageNotifications](#)
Minimum number of notifications needed for the current image settings on the camera.
- int [grabTimeout](#)
Time in milliseconds that [RetrieveBuffer\(\)](#) and [WaitForBufferEvent\(\)](#) will wait for an image before timing out and returning.
- [GrabMode](#) [grabMode](#)
Grab mode for the camera.
- [BusSpeed](#) [isochBusSpeed](#)
Isochronous bus speed.
- [BusSpeed](#) [asyncBusSpeed](#)
Asynchronous bus speed.
- [BandwidthAllocation](#) [bandwidthAllocation](#)
Bandwidth allocation flag that tells the camera the bandwidth allocation strategy to employ.
- unsigned int [registerTimeoutRetries](#)
Number of retries to perform when a register read/write timeout is received by the library.
- unsigned int [registerTimeout](#)
Register read/write timeout value, in microseconds.

7.10.1 Detailed Description

Configuration for a camera.

These options are options that are generally should be set before starting isochronous transfer.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 **FC2Config**() `[inline]`

7.10.3 Property Documentation

7.10.3.1 **BusSpeed** `asyncBusSpeed`

Asynchronous bus speed.

7.10.3.2 **BandwidthAllocation** `bandwidthAllocation`

Bandwidth allocation flag that tells the camera the bandwidth allocation strategy to employ.

7.10.3.3 **GrabMode** `grabMode`

Grab mode for the camera.

The default is DROP_FRAMES.

7.10.3.4 `int grabTimeout`

Time in milliseconds that `RetrieveBuffer()` and `WaitForBufferEvent()` will wait for an image before timing out and returning.

7.10.3.5 **BusSpeed** `isochBusSpeed`

Isochronous bus speed.

7.10.3.6 `unsigned int minNumImageNotifications`

Minimum number of notifications needed for the current image settings on the camera.
Read-only value.

7.10.3.7 `unsigned int numBuffers`

Number of buffers used by the [FlyCapture2](#) library to grab images.

7.10.3.8 `unsigned int numImageNotifications`

Number of notifications per image.

This value should only be set after the image settings to be used is set to the camera. The default number of notifications is 1.

There are 4 general scenarios:

- 1 notification - End of image
- 2 notifications - After first packet and end of image
- 3 notifications - After first packet, middle of image, end of image
- x notifications - After first packet, (x -2) spread evenly, end of image

7.10.3.9 unsigned int registerTimeout

Register read/write timeout value, in microseconds.

The default value is dependent on the interface type.

7.10.3.10 unsigned int registerTimeoutRetries

Number of retries to perform when a register read/write timeout is received by the library.

The default value is 0.

7.11 FC2Exception Class Reference

Exception that is thrown when an error is encountered.

Public Member Functions

- [FC2Exception](#) ()
- [FC2Exception](#) (String^ string)
- [FC2Exception](#) (String^ string, Exception^ exception)
- [~FC2Exception](#) ()

Protected Member Functions

- [FC2Exception](#) (Runtime::Serialization::SerializationInfo^ serializationInfo, - Runtime::Serialization::StreamingContext context)

Package Functions

- [FC2Exception](#) (FlyCapture2::Error error)

Properties

- [ErrorType Type](#) [get]
- [ErrorType CauseType](#) [get]
- [String^ NativeErrorTrace](#) [get]

7.11.1 Detailed Description

Exception that is thrown when an error is encountered.

This is used instead of returning an Error object as used in the C++ interface.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 FC2Exception ()

7.11.2.2 FC2Exception ([String^ string](#))

7.11.2.3 FC2Exception ([String^ string](#), [Exception^ exception](#))

7.11.2.4 ~FC2Exception ()

7.11.2.5 FC2Exception ([Runtime::Serialization::SerializationInfo^ serializationInfo](#), [Runtime::Serialization::StreamingContext context](#)) [protected]

7.11.2.6 FC2Exception ([FlyCapture2::Error error](#)) [package]

7.11.3 Property Documentation

7.11.3.1 [ErrorType CauseType](#) [get]

7.11.3.2 [String^ NativeErrorTrace](#) [get]

7.11.3.3 [ErrorType Type](#) [get]

7.12 FC2Version Struct Reference

The current version of the library.

Properties

- unsigned int [major](#)
Major version number.
- unsigned int [minor](#)
Minor version number.

- unsigned int [type](#)
Type version number.
- unsigned int [build](#)
Build version number.

7.12.1 Detailed Description

The current version of the library.

7.12.2 Property Documentation

7.12.2.1 unsigned int build

Build version number.

7.12.2.2 unsigned int major

Major version number.

7.12.2.3 unsigned int minor

Minor version number.

7.12.2.4 unsigned int type

Type version number.

7.13 Format7ImageSettings Struct Reference

Format 7 image settings.

Properties

- [Mode mode](#)
Format 7 mode.
- unsigned int [offsetX](#)
Horizontal image offset.
- unsigned int [offsetY](#)
Vertical image offset.
- unsigned int [width](#)
Width of image.

- unsigned int [height](#)
Height of image.
- [PixelFormat](#) [pixelFormat](#)
Pixel format of image.

7.13.1 Detailed Description

Format 7 image settings.

7.13.2 Property Documentation

7.13.2.1 unsigned int height

Height of image.

7.13.2.2 Mode mode

Format 7 mode.

7.13.2.3 unsigned int offsetX

Horizontal image offset.

7.13.2.4 unsigned int offsetY

Vertical image offset.

7.13.2.5 PixelFormat pixelFormat

Pixel format of image.

7.13.2.6 unsigned int width

Width of image.

7.14 Format7Info Struct Reference

Format 7 information for a single mode.

Properties

- [Mode mode](#)
Format 7 mode.
- unsigned int [maxWidth](#)
Maximum image width.
- unsigned int [maxHeight](#)
Maximum image height.
- unsigned int [offsetHStepSize](#)
Horizontal step size for the offset.
- unsigned int [offsetVStepSize](#)
Vertical step size for the offset.
- unsigned int [imageHStepSize](#)
Horizontal step size for the image.
- unsigned int [imageVStepSize](#)
Vertical step size for the image.
- unsigned int [pixelFormatBitField](#)
Supported pixel formats in a bit field.
- unsigned int [vendorPixelFormatBitField](#)
Vendor unique pixel formats in a bit field.
- unsigned int [packetSize](#)
Current packet size in bytes.
- unsigned int [minPacketSize](#)
Minimum packet size in bytes for current mode.
- unsigned int [maxPacketSize](#)
Maximum packet size in bytes for current mode.
- float [percentage](#)
Current packet size as a percentage of maximum packet size.

7.14.1 Detailed Description

Format 7 information for a single mode.

7.14.2 Property Documentation

7.14.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

7.14.2.2 unsigned int imageVStepSize

Vertical step size for the image.

7.14.2.3 unsigned int maxHeight

Maximum image height.

7.14.2.4 unsigned int maxPacketSize

Maximum packet size in bytes for current mode.

7.14.2.5 unsigned int maxWidth

Maximum image width.

7.14.2.6 unsigned int minPacketSize

Minimum packet size in bytes for current mode.

7.14.2.7 Mode mode

Format 7 mode.

7.14.2.8 unsigned int offsetHStepSize

Horizontal step size for the offset.

7.14.2.9 unsigned int offsetVStepSize

Vertical step size for the offset.

7.14.2.10 unsigned int packetSize

Current packet size in bytes.

7.14.2.11 float percentage

Current packet size as a percentage of maximum packet size.

7.14.2.12 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

7.14.2.13 unsigned int vendorPixelFormatBitField

Vendor unique pixel formats in a bit field.

7.15 Format7PacketInfo Struct Reference

Format 7 packet information.

Properties

- unsigned int [recommendedBytesPerPacket](#)
Recommended bytes per packet.
- unsigned int [maxBytesPerPacket](#)
Maximum bytes per packet.
- unsigned int [unitBytesPerPacket](#)
Minimum bytes per packet.

7.15.1 Detailed Description

Format 7 packet information.

7.15.2 Property Documentation

7.15.2.1 unsigned int maxBytesPerPacket

Maximum bytes per packet.

7.15.2.2 unsigned int recommendedBytesPerPacket

Recommended bytes per packet.

7.15.2.3 unsigned int unitBytesPerPacket

Minimum bytes per packet.

7.16 GigEConfig Struct Reference

Configuration for a GigE camera.

Properties

- bool [enablePacketResend](#)
Turn on/off packet resend functionality.

7.16.1 Detailed Description

Configuration for a GigE camera.

These options are options that are generally should be set before starting isochronous transfer.

7.16.2 Property Documentation

7.16.2.1 bool [enablePacketResend](#)

Turn on/off packet resend functionality.

7.17 GigEImageSettings Struct Reference

Image settings for a GigE camera.

Properties

- unsigned int [offsetX](#)
Horizontal image offset.
- unsigned int [offsetY](#)
Vertical image offset.
- unsigned int [width](#)
Width of image.
- unsigned int [height](#)
Height of image.
- [PixelFormat](#) [pixelFormat](#)
Pixel format of image.

7.17.1 Detailed Description

Image settings for a GigE camera.

7.17.2 Property Documentation

7.17.2.1 unsigned int height

Height of image.

7.17.2.2 unsigned int offsetX

Horizontal image offset.

7.17.2.3 unsigned int offsetY

Vertical image offset.

7.17.2.4 PixelFormat pixelFormat

Pixel format of image.

7.17.2.5 unsigned int width

Width of image.

7.18 GigImageSettingsInfo Struct Reference

Format 7 information for a single mode.

Properties

- unsigned int [maxWidth](#)
Maximum image width.
- unsigned int [maxHeight](#)
Maximum image height.
- unsigned int [offsetHStepSize](#)
Horizontal step size for the offset.
- unsigned int [offsetVStepSize](#)
Vertical step size for the offset.
- unsigned int [imageHStepSize](#)
Horizontal step size for the image.
- unsigned int [imageVStepSize](#)
Vertical step size for the image.
- unsigned int [pixelFormatBitField](#)
Supported pixel formats in a bit field.

- unsigned int [vendorPixelFormatBitField](#)
Vendor unique pixel formats in a bit field.

7.18.1 Detailed Description

Format 7 information for a single mode.

7.18.2 Property Documentation

7.18.2.1 unsigned int imageHStepSize

Horizontal step size for the image.

7.18.2.2 unsigned int imageVStepSize

Vertical step size for the image.

7.18.2.3 unsigned int maxHeight

Maximum image height.

7.18.2.4 unsigned int maxWidth

Maximum image width.

7.18.2.5 unsigned int offsetHStepSize

Horizontal step size for the offset.

7.18.2.6 unsigned int offsetVStepSize

Vertical step size for the offset.

7.18.2.7 unsigned int pixelFormatBitField

Supported pixel formats in a bit field.

7.18.2.8 unsigned int vendorPixelFormatBitField

Vendor unique pixel formats in a bit field.

7.19 GigEProperty Struct Reference

A GigE property.

Properties

- [GigEPropertyType propType](#)
The type of property.
- bool [isReadable](#)
Whether the property is readable.
- bool [isWritable](#)
Whether the property is writable.
- unsigned int [min](#)
Minimum value.
- unsigned int [max](#)
Maximum value.
- unsigned int [value](#)
Current value.

7.19.1 Detailed Description

A GigE property.

7.19.2 Property Documentation

7.19.2.1 bool isReadable

Whether the property is readable.

If this is false, then no other value in this structure is valid.

7.19.2.2 bool isWritable

Whether the property is writable.

7.19.2.3 unsigned int max

Maximum value.

7.19.2.4 unsigned int min

Minimum value.

7.19.2.5 GigEPropertyType propType

The type of property.

7.19.2.6 unsigned int value

Current value.

7.20 GigEStreamChannel Struct Reference

Information about a single GigE stream channel.

Properties

- unsigned int [networkInterfaceIndex](#)
Network interface index used (or to use).
- unsigned int [hostPort](#)
Host port on the PC where the camera will send the data stream.
- bool [doNotFragment](#)
Disable IP fragmentation of packets.
- unsigned int [packetSize](#)
Packet size, in bytes.
- unsigned int [interPacketDelay](#)
Inter packet delay, in timestamp counter units.
- System::Net::IPAddress^ [destinationIpAddress](#)
Destination IP address.
- unsigned int [sourcePort](#)
Source UDP port of the stream channel.

7.20.1 Detailed Description

Information about a single GigE stream channel.

7.20.2 Property Documentation

7.20.2.1 System::Net::IPAddress^ destinationIpAddress

Destination IP address.

It can be a multicast or unicast address.

7.20.2.2 bool doNotFragment

Disable IP fragmentation of packets.

7.20.2.3 unsigned int hostPost

Host port on the PC where the camera will send the data stream.

7.20.2.4 unsigned int interPacketDelay

Inter packet delay, in timestamp counter units.

7.20.2.5 unsigned int networkInterfaceIndex

Network interface index used (or to use).

7.20.2.6 unsigned int packetSize

Packet size, in bytes.

7.20.2.7 unsigned int sourcePort

Source UDP port of the stream channel.

Read only.

7.21 H264Option Struct Reference

Options for saving H.264 files.

Public Member Functions

- [H264Option](#) ()

Properties

- float [frameRate](#)
Frame rate of the stream.
- int [width](#)
Width of source image.
- int [height](#)
Height of source image.

- int [bitrate](#)

Bitrate to encode at.

7.21.1 Detailed Description

Options for saving H.264 files.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 H264Option () `[inline]`

7.21.3 Property Documentation

7.21.3.1 int bitrate

Bitrate to encode at.

7.21.3.2 float frameRate

Frame rate of the stream.

7.21.3.3 int height

Height of source image.

7.21.3.4 int width

Width of source image.

7.22 ImageMetadata Struct Reference

Metadata related to an image.

Properties

- unsigned int [embeddedTimeStamp](#)
Embedded timestamp.
- unsigned int [embeddedGain](#)
Embedded gain.
- unsigned int [embeddedShutter](#)
Embedded shutter.

- unsigned int [embeddedBrightness](#)
Embedded brightness.
- unsigned int [embeddedExposure](#)
Embedded exposure.
- unsigned int [embeddedWhiteBalance](#)
Embedded white balance.
- unsigned int [embeddedFrameCounter](#)
Embedded frame counter.
- unsigned int [embeddedStrobePattern](#)
Embedded strobe pattern.
- unsigned int [embeddedGPIOPinState](#)
Embedded GPIO pin state.
- unsigned int [embeddedROIPosition](#)
Embedded ROI position.

7.22.1 Detailed Description

Metadata related to an image.

7.22.2 Property Documentation

7.22.2.1 unsigned int [embeddedBrightness](#)

Embedded brightness.

7.22.2.2 unsigned int [embeddedExposure](#)

Embedded exposure.

7.22.2.3 unsigned int [embeddedFrameCounter](#)

Embedded frame counter.

7.22.2.4 unsigned int [embeddedGain](#)

Embedded gain.

7.22.2.5 unsigned int [embeddedGPIOPinState](#)

Embedded GPIO pin state.

7.22.2.6 unsigned int embeddedROIPosition

Embedded ROI position.

7.22.2.7 unsigned int embeddedShutter

Embedded shutter.

7.22.2.8 unsigned int embeddedStrobePattern

Embedded strobe pattern.

7.22.2.9 unsigned int embeddedTimeStamp

Embedded timestamp.

7.22.2.10 unsigned int embeddedWhiteBalance

Embedded white balance.

7.23 JpegOption Struct Reference

Options for saving JPEG image.

Public Member Functions

- [JpegOption](#) ()

Properties

- bool [progressive](#)
Whether to save as a progressive JPEG file.
- unsigned int [quality](#)
JPEG image quality in range (0-100).

7.23.1 Detailed Description

Options for saving JPEG image.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 JpegOption () [inline]

7.23.3 Property Documentation

7.23.3.1 bool progressive

Whether to save as a progressive JPEG file.

7.23.3.2 unsigned int quality

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

7.24 Jpg2Option Struct Reference

Options for saving JPEG2000 image.

Public Member Functions

- [Jpg2Option \(\)](#)

Properties

- unsigned int [quality](#)
JPEG saving quality in range (1-512).

7.24.1 Detailed Description

Options for saving JPEG2000 image.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 Jpg2Option () [inline]

7.24.3 Property Documentation

7.24.3.1 unsigned int quality

JPEG saving quality in range (1-512).

7.25 LutData Struct Reference

Information about the camera's look up table.

Properties

- bool [supported](#)
Flag indicating if LUT is supported.
- bool [enabled](#)
Flag indicating if LUT is enabled.
- unsigned int [numBanks](#)
The number of LUT banks available (Always 1 for PGR LUT).
- unsigned int [numChannels](#)
The number of LUT channels per bank available.
- unsigned int [inputBitDepth](#)
The input bit depth of the LUT.
- unsigned int [outputBitDepth](#)
The output bit depth of the LUT.
- unsigned int [numEntries](#)
The number of entries in the LUT.

7.25.1 Detailed Description

Information about the camera's look up table.

7.25.2 Property Documentation

7.25.2.1 bool enabled

Flag indicating if LUT is enabled.

7.25.2.2 unsigned int inputBitDepth

The input bit depth of the LUT.

7.25.2.3 unsigned int numBanks

The number of LUT banks available (Always 1 for PGR LUT).

7.25.2.4 unsigned int numChannels

The number of LUT channels per bank available.

7.25.2.5 unsigned int numEntries

The number of entries in the LUT.

7.25.2.6 unsigned int outputBitDepth

The output bit depth of the LUT.

7.25.2.7 bool supported

Flag indicating if LUT is supported.

7.26 ManagedAVIRecorder Class Reference

[ManagedAVIRecorder](#) provides the functionality for the user to record images to an AVI file.

Public Member Functions

- [ManagedAVIRecorder](#) ()
- [~ManagedAVIRecorder](#) ()
- void [AVIOpen](#) (System::String^ fileName, [AviOption](#)^ option)
Open an AVI file in preparation for writing Images to disk.
- void [AVIOpen](#) (System::String^ fileName, [MJPGOption](#)^ option)
Open an MJPEG AVI file in preparation for writing Images to disk.
- void [AVIOpen](#) (System::String^ fileName, [H264Option](#)^ option)
Open an H.264 MP4 file in preparation for writing Images to disk.
- void [AVIAppend](#) ([ManagedImage](#)^ image)
Append an image to the AVI file.
- void [AVIClose](#) ()
Close the AVI file.

7.26.1 Detailed Description

[ManagedAVIRecorder](#) provides the functionality for the user to record images to an AVI file.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 [ManagedAVIRecorder](#) ()

7.26.2.2 [~ManagedAVIRecorder](#) ()

7.26.3 Member Function Documentation

7.26.3.1 void [AVIAppend](#) ([ManagedImage](#)[^] *image*)

Append an image to the AVI file.

Parameters

<i>image</i>	The ManagedImage to append.
--------------	---

7.26.3.2 void [AVIClose](#) ()

Close the AVI file.

See also

[AVIOpen\(\)](#)

7.26.3.3 void [AVIOpen](#) ([System::String](#)[^] *fileName*, [AviOption](#)[^] *option*)

Open an AVI file in preparation for writing Images to disk.

The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>fileName</i>	The filename of the AVI file.
<i>option</i>	Options to apply to the AVI file.

See also

[AVIClose\(\)](#)

7.26.3.4 void AVIOpen (System::String^ *fileName*, MJPGOption^ *option*)

Open an MJPEG AVI file in preparation for writing Images to disk.

The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>fileName</i>	The filename of the AVI file.
<i>option</i>	Options to apply to the AVI file.

See also

[AVIClose\(\)](#)

7.26.3.5 void AVIOpen (System::String^ *fileName*, H264Option^ *option*)

Open an H.264 MP4 file in preparation for writing Images to disk.

The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>fileName</i>	The filename of the AVI file.
<i>option</i>	Options to apply to the AVI file.

See also

[AVIClose\(\)](#)

7.27 ManagedBusManager Class Reference

[ManagedBusManager](#) provides the functionality for the user to get an PGRGuid for a desired camera or device easily.

Public Member Functions

- [ManagedBusManager](#) ()
- [~ManagedBusManager](#) ()
- void [FireBusReset](#) ([ManagedPGRGuid](#)^ guid)
Fire a bus reset.
- unsigned int [GetNumOfCameras](#) ()
Gets the number of cameras attached to the PC.

- [ManagedPGRGuid](#)[^] [GetCameraFromIPAddress](#) (System::Net::IPAddress[^] ipAddress)
Gets the [ManagedPGRGuid](#) for a camera with the specified IPv4 address.
- [ManagedPGRGuid](#)[^] [GetCameraFromIndex](#) (unsigned int index)
Gets the [ManagedPGRGuid](#) for a camera on the PC.
- [ManagedPGRGuid](#)[^] [GetCameraFromSerialNumber](#) (unsigned int serialNumber)
Gets the [ManagedPGRGuid](#) for a camera on the PC.
- unsigned int [GetCameraSerialNumberFromIndex](#) (unsigned int index)
Gets the serial number of the camera with the specified index.
- [InterfaceType](#) [GetInterfaceTypeFromGuid](#) ([ManagedPGRGuid](#)[^] guid)
Gets the interface type associated with a [ManagedPGRGuid](#).
- unsigned int [GetNumOfDevices](#) ()
Gets the number of devices.
- [ManagedPGRGuid](#)[^] [GetDeviceFromIndex](#) (unsigned int index)
Gets the [ManagedPGRGuid](#) for a device.
- unsigned int [ReadPhyRegister](#) ([ManagedPGRGuid](#)[^] guid, unsigned int page, unsigned int port, unsigned int address)
Read a phy register on the specified device.
- void [WritePhyRegister](#) ([ManagedPGRGuid](#)[^] guid, unsigned int page, unsigned int port, unsigned int address, unsigned int regVal)
Write a phy register on the specified device.
- unsigned int [GetUsbLinkInfo](#) ([ManagedPGRGuid](#)[^] guid)
Read usb link info for the port that the specified device is connected to.
- unsigned int [GetUsbPortStatus](#) ([ManagedPGRGuid](#)[^] guid)
Read usb port status for the port that the specified device is connected to.
- [ManagedTopologyNode](#)[^] [GetTopology](#) ()
Gets the topology information for the PC.
- void [RescanBus](#) ()
Force a rescan of the buses.
- System::IntPtr [RegisterCallback](#) (EnumCallback[^] hCallbackDelegate, [ManagedCallbackType](#) callbackType, System::IntPtr parameter)
Register a callback function that will be called when the specified callback event occurs.
- void [UnregisterCallback](#) (System::IntPtr callbackHandle)
Unregister a callback function.
- bool [IsCameraControlable](#) ([ManagedPGRGuid](#)[^] guid)
Query CCP status on camera with corresponding PGRGuid.

Static Public Member Functions

- static void [ForceIPAddressToCamera](#) (System::Net::NetworkInformation::PhysicalAddress^ macAddress, System::Net::IPAddress^ ipAddress, System::Net::IPAddress^ subnetMask, System::Net::IPAddress^ defaultGateway)
Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.
- static void [ForceAllIPAddressesAutomatically](#) ()
Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the network adapters that they are connected to.
- static void [ForceAllIPAddressesAutomatically](#) (unsigned int serialNumber)
Force a specific cameras on the network to be assigned sequential IP address on the same subnet as the network adapters that it is connected to.
- static array< [CameraInfo](#)^ >^ [DiscoverGigECameras](#) ()
Discover all cameras connected to the network even if they reside on a different subnet.

Protected Member Functions

- [IManagedBusManager](#) ()

Static Package Functions

- static void [ConvertToNativeGuid](#) ([ManagedPGRGuid](#)^ mgdPGRGuid, FlyCapture2::PGRGuid *pgrGuid)
Convert a [ManagedPGRGuid](#) to a native PGRGuid.
- static void [ConvertToManagedGuid](#) (FlyCapture2::PGRGuid *pgrGuid, [ManagedPGRGuid](#)^ mgdPGRGuid)
Convert a native PGRGuid to a [ManagedPGRGuid](#).

7.27.1 Detailed Description

[ManagedBusManager](#) provides the functionality for the user to get an PGRGuid for a desired camera or device easily.

Once the camera or device token is found, it can then be used to connect to the camera or device through the camera class or device class. In addition, the BusManager class provides the ability to be notified when a camera or device is added or removed or some event occurs on the PC.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 ManagedBusManager ()

7.27.2.2 ~ManagedBusManager ()

7.27.2.3 `!ManagedBusManager ()` [protected]

7.27.3 Member Function Documentation

7.27.3.1 `void ConvertToManagedGuid (FlyCapture2::PGRGuid * pgrGuid, ManagedPGRGuid^ mgdPGRGuid)` [inline, static, package]

Convert a native PGRGuid to a [ManagedPGRGuid](#).

Parameters

<i>pgrGuid</i>	The native PGRGuid.
<i>mgdPGR-Guid</i>	The ManagedPGRGuid .

7.27.3.2 `void ConvertToNativeGuid (ManagedPGRGuid^ mgdPGRGuid, FlyCapture2::PGRGuid * pgrGuid)` [inline, static, package]

Convert a [ManagedPGRGuid](#) to a native PGRGuid.

Parameters

<i>mgdPGR-Guid</i>	The ManagedPGRGuid .
<i>pgrGuid</i>	The native PGRGuid.

7.27.3.3 `array< CameraInfo^ > DiscoverGigECameras ()` [static]

Discover all cameras connected to the network even if they reside on a different subnet.

This is useful in situations where a GigE camera is using Persistent IP and the application's subnet is different from the device subnet. After discovering the camera, it is easy to use [ForceIPAddressToCamera\(\)](#) to set a different IP configuration.

Returns

Array of [CameraInfo](#) structures containing information about discovered cameras.

7.27.3.4 `void FireBusReset (ManagedPGRGuid^ guid)`

Fire a bus reset.

The actual bus reset is only fired for the specified 1394 bus, but it will effectively cause a global bus reset for the library.

Parameters

<i>guid</i>	ManagedPGRGuid of the camera or the device to cause bus reset.
-------------	--

7.27.3.5 void ForceAllIPAddressesAutomatically () [static]

Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the network adapters that they are connected to.

This is useful in situations where a GigE Vision cameras are using Persistent IP addresses and the application's subnet is different from the devices.

7.27.3.6 void ForceAllIPAddressesAutomatically (unsigned int *serialNumber*) [static]

Force a specific cameras on the network to be assigned sequential IP address on the same subnet as the network adapters that it is connected to.

This is useful in situations where a GigE Vision camera is using Persistent IP addresses and the application's subnet is different from the device.

7.27.3.7 void ForceIPAddressToCamera (System::Net::NetworkInformation::PhysicalAddress^ *macAddress*, System::Net::IPAddress^ *ipAddress*, System::Net::IPAddress^ *subnetMask*, System::Net::IPAddress^ *defaultGateway*) [static]

Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.

This is useful in situations where a GigE Vision camera is using Persistent IP and the application's subnet is different from the device subnet.

Parameters

<i>macAddress</i>	MAC address of the camera.
<i>ipAddress</i>	IP address to set on the camera.
<i>subnetMask</i>	Subnet mask to set on the camera.
<i>default-Gateway</i>	Default gateway to set on the camera.

7.27.3.8 ManagedPGRGuid GetCameraFromIndex (unsigned int *index*)

Gets the [ManagedPGRGuid](#) for a camera on the PC.

It uniquely identifies the camera specified by the index and is used to identify the camera during a [ManagedCamera::Connect\(\)](#) call.

Parameters

<i>index</i>	Zero based index of camera.
--------------	-----------------------------

Returns

Unique [ManagedPGRGuid](#) for the camera.

**7.27.3.9 ManagedPGRGuid GetCameraFromIPAddress (System::Net::IPAddress^
ipAddress)**

Gets the [ManagedPGRGuid](#) for a camera with the specified IPv4 address.

Parameters

<i>ipAddress</i>	IP address to get ManagedPGRGuid for.
------------------	---

Returns

Unique [ManagedPGRGuid](#) for the camera.

7.27.3.10 ManagedPGRGuid GetCameraFromSerialNumber (unsigned int serialNumber)

Gets the [ManagedPGRGuid](#) for a camera on the PC.

It uniquely identifies the camera specified by the serial number and is used to identify the camera during a [ManagedCamera::Connect\(\)](#) call.

Parameters

<i>serial- Number</i>	Serial number of camera.
---------------------------	--------------------------

See also

[GetCameraFromIndex\(\)](#)

Returns

Unique [ManagedPGRGuid](#) for the camera.

7.27.3.11 unsigned int GetCameraSerialNumberFromIndex (unsigned int index)

Gets the serial number of the camera with the specified index.

Parameters

<i>index</i>	Zero based index of desired camera.
--------------	-------------------------------------

Returns

Serial number of camera.

7.27.3.12 ManagedPGRGuid GetDeviceFromIndex (unsigned int *index*)

Gets the [ManagedPGRGuid](#) for a device.

It uniquely identifies the device specified by the index.

Parameters

<i>index</i>	Zero based index of device.
--------------	-----------------------------

See also

[GetNumOfDevices\(\)](#)

Returns

Unique [ManagedPGRGuid](#) for the device.

7.27.3.13 InterfaceType GetInterfaceTypeFromGuid (ManagedPGRGuid^ *guid*)

Gets the interface type associated with a [ManagedPGRGuid](#).

This is useful in situations where there is a need to enumerate all cameras for a particular interface.

Parameters

<i>guid</i>	The ManagedPGRGuid to get the interface for.
-------------	--

Returns

The interface type of the PGRGuid.

7.27.3.14 unsigned int GetNumOfCameras ()

Gets the number of cameras attached to the PC.

Returns

The number of cameras attached.

7.27.3.15 unsigned int GetNumOfDevices ()

Gets the number of devices.

This may include hubs, host controllers and other hardware devices (including cameras).

Returns

The number of devices found.

7.27.3.16 ManagedTopologyNode GetTopology ()

Gets the topology information for the PC.

Returns

[ManagedTopologyNode](#) object that will contain the topology

7.27.3.17 unsigned int GetUsbLinkInfo (ManagedPGRGuid^ *guid*)

Read usb link info for the port that the specified device is connected to.

Parameters

<i>guid</i>	PGRGuid of the device to read from.
-------------	-------------------------------------

Returns

Value read from the card register.

7.27.3.18 unsigned int GetUsbPortStatus (ManagedPGRGuid^ *guid*)

Read usb port status for the port that the specified device is connected to.

Parameters

<i>guid</i>	PGRGuid of the device to read from.
-------------	-------------------------------------

Returns

Value read from the card register.

7.27.3.19 bool IsCameraControlable (ManagedPGRGuid^ *guid*)

Query CCP status on camera with corresponding PGRGuid.

This is useful to determine if a GigE camera can be controlled.

Parameters

<i>pGuid</i>	PGRGuid of the camera
--------------	-----------------------

Returns

True means camera is controllable.

7.27.3.20 `unsigned int ReadPhyRegister (ManagedPGRGuid^ guid, unsigned int page, unsigned int port, unsigned int address)`

Read a phy register on the specified device.

The full address to be read from is determined by the page, port and address.

Parameters

<i>guid</i>	ManagedPGRGuid of the device to read from.
<i>page</i>	Page to read from.
<i>port</i>	Port to read from.
<i>address</i>	Address to read from.

Returns

Value read from the phy register.

7.27.3.21 `System::IntPtr RegisterCallback (EnumCallback^ hCallbackDelegate, ManagedCallbackType callbackType, System::IntPtr parameter)`

Register a callback function that will be called when the specified callback event occurs.

Parameters

<i>hCallback-Delegate</i>	Handle to EnumCallback function to receive the callback.
<i>callbackType</i>	Type of callback to register for.
<i>parameter</i>	Callback parameter to be passed to callback.

See also

[UnregisterCallback\(\)](#)

Returns

Unique callback handle used for unregistering callback.

7.27.3.22 void RescanBus ()

Force a rescan of the buses.

This does not trigger a bus reset. However, any current connections to a [Managed-Camera](#) object will be invalidated.

7.27.3.23 void UnregisterCallback (System::IntPtr *callbackHandle*)

Unregister a callback function.

Parameters

<i>callback-Handle</i>	Unique callback handle.
------------------------	-------------------------

See also

[RegisterCallback\(\)](#)

7.27.3.24 void WritePhyRegister (ManagedPGRGuid^ *guid*, unsigned int *page*, unsigned int *port*, unsigned int *address*, unsigned int *regVal*)

Write a phy register on the specified device.

The full address to be written to is determined by the page, port and address.

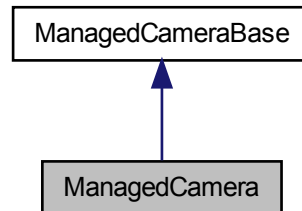
Parameters

<i>guid</i>	ManagedPGRGuid of the device to write to.
<i>page</i>	Page to write to.
<i>port</i>	Port to write to.
<i>address</i>	Address to write to.
<i>regVal</i>	Value to write to phy register.

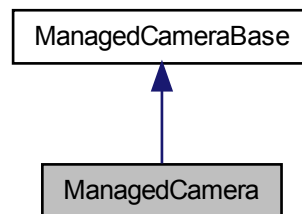
7.28 ManagedCamera Class Reference

[ManagedCamera](#) represents a physical camera that uses the IIDC register set.

Inheritance diagram for ManagedCamera:



Collaboration diagram for ManagedCamera:



Public Member Functions

- [ManagedCamera](#) ()
- [~ManagedCamera](#) ()
- virtual void [Connect](#) ([ManagedPGRGuid](#)^ mgdPGRGuid) override
Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Protected Member Functions

- [!ManagedCamera](#) ()

DCAM Formats

These functions deal with DCAM video mode and frame rate on the camera.

- bool [GetVideoModeAndFrameRateInfo](#) ([VideoMode](#) videoMode, [FrameRate](#) frameRate)
Query the camera to determine if the specified video mode and frame rate is supported.
- void [GetVideoModeAndFrameRate](#) ([VideoMode](#)% videoMode, [FrameRate](#)% frameRate)
Get the current video mode and frame rate from the camera.
- void [SetVideoModeAndFrameRate](#) ([VideoMode](#) videoMode, [FrameRate](#) frameRate)
Set the specified video mode and frame rate to the camera.

Format7

These functions deal with Format7 custom image control on the camera.

- [Format7Info](#)^ [GetFormat7Info](#) ([Mode](#) mode, bool% supported)
Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.
- [Format7PacketInfo](#)^ [ValidateFormat7Settings](#) ([Format7ImageSettings](#)^ imageSettings, bool% settingsAreValid)
Validates [Format7ImageSettings](#) structure and returns valid packet size information if the image settings are valid.
- void [GetFormat7Configuration](#) ([Format7ImageSettings](#)^ imageSettings, unsigned int% packetSize, float% percentSpeed)
Get the current Format7 configuration from the camera.
- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)^ imageSettings, unsigned int recommendedPacketSize)
Set the current Format7 configuration to the camera.
- void [SetFormat7Configuration](#) ([Format7ImageSettings](#)^ imageSettings, float recommendedPercentSpeed)
Set the current Format7 configuration to the camera.
- static void [StartSyncCapture](#) (unsigned int numCameras, array< [Managed-Camera](#)^ >^ppCameras)
Start multiple cameras in synchronization.
- static void [StartSyncCapture](#) (unsigned int numCameras, array< [Managed-Camera](#)^ >^ppCameras, array< [ImageEventCallback](#)^ >^pCallbackFns, array< [IntPtr](#)^ >^pCallbackDataArray)
Start multiple cameras in synchronization using callbacks.

7.28.1 Detailed Description

[ManagedCamera](#) represents a physical camera that uses the IIDC register set.

The object must first be connected to using [Connect\(\)](#) before any other operations can proceed.

It is possible for more than 1 Camera object to connect to a single physical camera. However, isochronous transmission to more than 1 Camera object is not supported.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 ManagedCamera ()

7.28.2.2 ~ManagedCamera ()

7.28.2.3 !ManagedCamera () [protected]

7.28.3 Member Function Documentation

7.28.3.1 void Connect (ManagedPGRGuid^ *mgdPGRGuid*) [override, virtual]

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

7.28.3.2 void GetFormat7Configuration (Format7ImageSettings^ *imageSettings*, unsigned int% *packetSize*, float% *percentSpeed*)

Get the current Format7 configuration from the camera.

This call will only succeed if the camera is already in Format7.

Parameters

<i>image-Settings</i>	Current image settings.
<i>packetSize</i>	Current packet size.
<i>percent-Speed</i>	Current packet size as a percentage.

See also

[GetFormat7Info\(\)](#)
[ValidateFormat7Settings\(\)](#)
[SetFormat7Configuration\(\)](#)
[GetVideoModeAndFrameRate\(\)](#)

7.28.3.3 **Format7Info** GetFormat7Info (**Mode** *mode*, **bool%** *supported*)

Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.

The mode must be specified in the [Format7Info](#) structure in order for the function to succeed.

Parameters

<i>mode</i>	Format7 mode to query.
<i>supported</i>	Whether the specified mode is supported.

See also

[ValidateFormat7Settings\(\)](#)
[GetFormat7Configuration\(\)](#)
[SetFormat7Configuration\(\)](#)

Returns

[Format7Info](#) structure filled with the capabilities of the specified mode and the current state in the specified mode.

7.28.3.4 **void** GetVideoModeAndFrameRate (**VideoMode%** *videoMode*, **FrameRate%** *frameRate*)

Get the current video mode and frame rate from the camera.

If the camera is in Format7, the video mode will be VIDEOMODE_FORMAT7 and the frame rate will be FRAMERATE_FORMAT7.

Parameters

<i>videoMode</i>	Current video mode.
<i>frameRate</i>	Current frame rate.

See also

[GetVideoModeAndFrameRateInfo\(\)](#)
[SetVideoModeAndFrameRate\(\)](#)

7.28.3.5 **bool** GetVideoModeAndFrameRateInfo (**VideoMode** *videoMode*, **FrameRate** *frameRate*)

Query the camera to determine if the specified video mode and frame rate is supported.

Parameters

<i>videoMode</i>	Video mode to check.
<i>frameRate</i>	Frame rate to check.

See also

[GetVideoModeAndFrameRate\(\)](#)

[SetVideoModeAndFrameRate\(\)](#)

Returns

Whether the video mode and frame rate is supported.

7.28.3.6 **void** SetFormat7Configuration (**Format7ImageSettings**[^] *imageSettings*, unsigned int *recommendedPacketSize*)

Set the current Format7 configuration to the camera.

Parameters

<i>image-Settings</i>	Image settings to be written to the camera.
<i>recommended-PacketSize</i>	Packet size to be written to the camera.

See also

[GetFormat7Info\(\)](#)

[ValidateFormat7Settings\(\)](#)

[GetFormat7Configuration\(\)](#)

7.28.3.7 **void** SetFormat7Configuration (**Format7ImageSettings**[^] *imageSettings*, float *recommendedPercentSpeed*)

Set the current Format7 configuration to the camera.

Parameters

<i>image-Settings</i>	Image settings to be written to the camera.
-----------------------	---

<i>recommended- Percent- Speed</i>	Percentage of packet size to be written to the camera.
--	--

See also

[GetFormat7Info\(\)](#)
[ValidateFormat7Settings\(\)](#)
[GetFormat7Configuration\(\)](#)

7.28.3.8 void SetVideoModeAndFrameRate (VideoMode *videoMode*, FrameRate *frameRate*)

Set the specified video mode and frame rate to the camera.

It is not possible to set the camera to VIDEOMODE_FORMAT7 or FRAMERATE_FORMAT7. Use the Format7 functions to set the camera into Format7.

Parameters

<i>videoMode</i>	Video mode to set to camera.
<i>frameRate</i>	Frame rate to set to camera.

See also

[GetVideoModeAndFrameRateInfo\(\)](#)
[GetVideoModeAndFrameRate\(\)](#)

7.28.3.9 void StartSyncCapture (unsigned int *numCameras*, array< ManagedCamera[^]> *ppCameras*) [static]

Start multiple cameras in synchronization.

Parameters

<i>numCameras</i>	Number of cameras to start.
<i>ppCameras</i>	An array of ManagedCamera objects to be started.

See also

[StartCapture\(\)](#)

7.28.3.10 `void StartSyncCapture (unsigned int numCameras, array< ManagedCamera^> ppCameras, array< ImageEventCallback^> pCallbackFns, array< IntPtr^> pCallbackDataArray) [static]`

Start multiple cameras in synchronization using callbacks.

Parameters

<i>num-Cameras</i>	Number of cameras to start.
<i>ppCameras</i>	An array of ManagedCamera objects to be started
<i>pCallback-Fns</i>	An array of callback functions
<i>pCallback-DataArray</i>	An array of ManagedImage objects to be populated during callback

See also

[StartCapture\(\)](#)

7.28.3.11 `Format7PacketInfo ValidateFormat7Settings (Format7ImageSettings^ imageSettings, bool% settingsAreValid)`

Validates [Format7ImageSettings](#) structure and returns valid packet size information if the image settings are valid.

The current image settings are cached while validation is taking place. The cached settings are restored when validation is complete.

Parameters

<i>image-Settings</i>	Structure containing the image settings.
<i>settingsAre-Valid</i>	Whether the settings are valid.

See also

[GetFormat7Info\(\)](#)
[GetFormat7Configuration\(\)](#)
[SetFormat7Configuration\(\)](#)

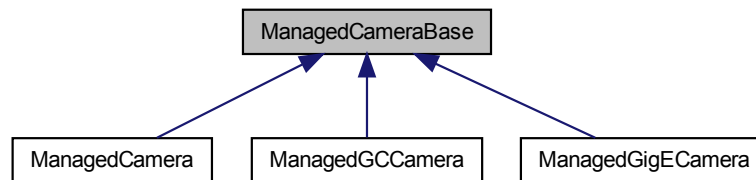
Returns

Packet size information that can be used to determine a valid packet size.

7.29 ManagedCameraBase Class Reference

Abstract base class that represents a generic camera that defines a general interface to a camera.

Inheritance diagram for ManagedCameraBase:

**Public Member Functions**

- virtual `~ManagedCameraBase ()`
- void `SetCamera (System::IntPtr otherCamera)`
Set camera from a integer pointer camera.
- virtual `TimeStamp^ GetCycleTime ()`
Returns a Timestamp struct containing 1394 CYCLE_TIME information.

Connection and Image Retrieval

These functions deal with connections and image retrieval from the camera.

- virtual void `Connect (ManagedPGRGuid^ mgdPGRGuid)`
Connects the ManagedCamera object to the camera specified by the GUID.
- virtual void `Disconnect ()`
Disconnects the ManagedCamera object from the camera.
- virtual bool `IsConnected ()`
Checks if the ManagedCamera object is currently connected to a physical camera.
- virtual void `SetCallback (ImageEventCallback^ hCallbackDelegate)`
Sets the callback data to be used on completion of image transfer.
- virtual void `StartCapture ()`
Starts isochronous image capture.
- virtual void `StartCapture (ImageEventCallback^ hCallbackDelegate)`
Starts isochronous image capture.

- virtual void [StopCapture](#) ()
Stops isochronous image transfer and cleans up all associated resources.
- virtual void [RetrieveBuffer](#) ([ManagedImage](#)^ image)
Retrieves the the next image object containing the next image.
- virtual void [WaitForBufferEvent](#) ([ManagedImage](#)^ image, unsigned int event-Number)
Retrieves the next image event containing the next part of the image.
- virtual [FC2Config](#)^ [GetConfiguration](#) ()
Get the configuration associated with the camera object.
- virtual void [SetConfiguration](#) ([FC2Config](#)^ config)
Set the configuration associated with the camera object.

Information and Properties

These functions deal with information and properties can be retrieved from the camera.

- virtual [CameraInfo](#)^ [GetCameraInfo](#) ()
Retrieves information from the camera such as serial number, model name and other camera information.
- virtual [CameraPropertyInfo](#)^ [GetPropertyInfo](#) ([PropertyType](#) type)
Retrieves information about the specified camera property.
- virtual [CameraProperty](#)^ [GetProperty](#) ([PropertyType](#) type)
Reads the settings for the specified property from the camera.
- virtual void [SetProperty](#) ([CameraProperty](#)^ camProperty)
Writes the settings for the specified property to the camera.
- virtual void [SetProperty](#) ([CameraProperty](#)^ camProperty, bool broadcast)
Writes the settings for the specified property to the camera.

General Purpose Input / Output

These functions deal with general GPIO pin control on the camera.

- virtual unsigned int [GetGPIOPinDirection](#) (unsigned int pin)
Get the GPIO pin direction for the specified pin.
- virtual void [SetGPIOPinDirection](#) (unsigned int pin, unsigned int direction)
Set the GPIO pin direction for the specified pin.
- virtual void [SetGPIOPinDirection](#) (unsigned int pin, unsigned int direction, bool broadcast)
Set the GPIO pin direction for the specified pin.

Trigger

These functions deal with trigger control on the camera.

- virtual [TriggerModelInfo](#)^ [GetTriggerModelInfo](#) ()
Retrieve trigger information from the camera.
- virtual [TriggerMode](#)^ [GetTriggerMode](#) ()
Retrieve current trigger settings from the camera.
- virtual void [SetTriggerMode](#) ([TriggerMode](#)^ triggerMode)
Set the specified trigger settings to the camera.
- virtual void [FireSoftwareTrigger](#) (bool broadcast)
Fire the software trigger according to the DCAM specifications.

Strobe

These functions deal with strobe control on the camera.

- virtual [StrobeInfo](#)[^] [GetStrobeInfo](#) (unsigned int source)
Retrieve strobe information from the camera.
- virtual [StrobeControl](#)[^] [GetStrobe](#) (unsigned int source)
Retrieve current strobe settings from the camera.
- virtual void [SetStrobe](#) ([StrobeControl](#)[^] strobeControl)
Set current strobe settings to the camera.

Look Up Table

These functions deal with Look Up Table control on the camera.

Note that some cameras may report support for the LUT and return an inputBitDepth of 0. In these cases use log2(numEntries) for the inputBitDepth.

- virtual [LutData](#)[^] [GetLUTInfo](#) ()
Query if LUT support is available on the camera.
- virtual void [GetLUTBankInfo](#) (unsigned int bank, bool% readSupported, bool% writeSupported)
Query the read/write status of a single LUT bank.
- virtual unsigned int [GetActiveLUTBank](#) ()
Get the LUT bank that is currently being used.
- virtual void [SetActiveLUTBank](#) (unsigned int activeBank)
Set the LUT bank that will be used.
- virtual void [EnableLUT](#) (bool on)
Enable or disable LUT functionality on the camera.
- virtual void [GetLUTChannel](#) (unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >[^]entries)
Get the LUT channel settings from the camera.
- virtual void [SetLUTChannel](#) (unsigned int bank, unsigned int channel, unsigned int sizeEntries, array< unsigned int >[^]entries)
Set the LUT channel settings to the camera.

Memory Channels

These functions deal with memory channel control on the camera.

- virtual unsigned int [GetMemoryChannel](#) ()
Retrieve the current memory channel from the camera.
- virtual void [SaveToMemoryChannel](#) (unsigned int channel)
Save the current settings to the specified current memory channel.
- virtual void [RestoreFromMemoryChannel](#) (unsigned int channel)
Restore the specified current memory channel.
- virtual unsigned int [GetMemoryChannelInfo](#) ()
Query the camera for memory channel support.

Embedded Image Information

These functions deal with embedded image information control on the camera.

- virtual [EmbeddedImageInfo](#)^ [GetEmbeddedImageInfo](#) ()
Get the current status of the embedded image information register, as well as the availability of each embedded property.
- virtual void [SetEmbeddedImageInfo](#) ([EmbeddedImageInfo](#)^ info)
Sets the on/off values of the embedded image information structure to the camera.

Register Operation

These functions deal with register operation on the camera.

- virtual void [WriteRegister](#) (unsigned int address, unsigned int value)
Write to the specified register on the camera.
- virtual void [WriteRegister](#) (unsigned int address, unsigned int value, bool broadcast)
Write to the specified register on the camera.
- virtual unsigned int [ReadRegister](#) (unsigned int address)
Read the specified register from the camera.
- virtual void [WriteRegisterBlock](#) (unsigned short addressHigh, unsigned int addressLow, array< unsigned int >^buffer)
Write to the specified register block on the camera.
- virtual void [ReadRegisterBlock](#) (unsigned short addressHigh, unsigned int addressLow, array< unsigned int >^buffer)
Read from the specified register block on the camera.

Static Public Member Functions

- static System::String^ [GetRegisterString](#) (unsigned int registerVal)
Returns a text representation of the register value.

Protected Member Functions

- [ManagedCameraBase](#) ()
- void [OnNativeCallback](#) (FlyCapture2::Image *pImage, void *pCallbackData)

Protected Attributes

- FlyCapture2::CameraBase * [m_pNativeCamBase](#)
- bool [m_isLocal](#)
- ImageEventCallback^ [m_externalDelegate](#)
- ImageCallbackDelegate^ [m_internalDelegate](#)

Package Functions

- FlyCapture2::CameraBase * [GetNativeCamera](#) ()

7.29.1 Detailed Description

Abstract base class that represents a generic camera that defines a general interface to a camera.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 `virtual ~ManagedCameraBase () [inline, virtual]`

7.29.2.2 `ManagedCameraBase () [inline, protected]`

7.29.3 Member Function Documentation

7.29.3.1 `void Connect (ManagedPGRGuid^ mgdPGRGuid) [virtual]`

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented in [ManagedCamera](#), [ManagedGigECamera](#), and [ManagedGCCamera](#).

7.29.3.2 `void Disconnect (void) [virtual]`

Disconnects the [ManagedCamera](#) object from the camera.

This allows another physical camera to be connected to the [ManagedCamera](#) object.

See also

[Connect\(\)](#)

Reimplemented in [ManagedGCCamera](#).

7.29.3.3 `void EnableLUT (bool on) [virtual]`

Enable or disable LUT functionality on the camera.

Parameters

<i>on</i>	Whether to enable or disable LUT.
-----------	-----------------------------------

See also

[GetLUTInfo\(\)](#)
[GetLUTChannel\(\)](#)
[SetLUTChannel\(\)](#)

7.29.3.4 void FireSoftwareTrigger (bool *broadcast*) [virtual]

Fire the software trigger according to the DCAM specifications.

Parameters

<i>broadcast</i>	Whether the action should be broadcast.
------------------	---

7.29.3.5 unsigned int GetActiveLUTBank () [virtual]

Get the LUT bank that is currently being used.

For cameras with PGR LUT, the active bank is always 0.

Returns

The currently active bank.

7.29.3.6 CameraInfo GetCameraInfo () [virtual]

Retrieves information from the camera such as serial number, model name and other camera information.

Returns

Structure containing camera information.

7.29.3.7 FC2Config GetConfiguration () [virtual]

Get the configuration associated with the camera object.

See also

[SetConfiguration\(\)](#)

Returns

Current configuration.

7.29.3.8 **TimeStamp** GetCycleTime () [virtual]

Returns a Timestamp struct containing 1394 CYCLE_TIME information.

Parameters

<i>registerVal</i>	The register value to query.
--------------------	------------------------------

Returns

An Error indicating the success or failure of the function.

7.29.3.9 **EmbeddedImageInfo** GetEmbeddedImageInfo () [virtual]

Get the current status of the embedded image information register, as well as the availability of each embedded property.

See also

[SetEmbeddedImageInfo\(\)](#)

Returns

[EmbeddedImageInfo](#) structure containing embedded image information.

7.29.3.10 **unsigned int** GetGPiOPinDirection (**unsigned int** *pin*) [virtual]

Get the GPIO pin direction for the specified pin.

This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters

<i>pin</i>	Pin to get the direction for.
------------	-------------------------------

See also

[SetGPiOPinDirection\(\)](#)

Returns

Direction of the pin. 0 for input, 1 for output.

7.29.3.11 void GetLUTBankInfo (unsigned int *bank*, bool% *readSupported*, bool% *writeSupported*) [virtual]

Query the read/write status of a single LUT bank.

Parameters

<i>bank</i>	The bank to query.
<i>read-Supported</i>	Whether reading from the bank is supported.
<i>write-Supported</i>	Whether writing to the bank is supported.

7.29.3.12 void GetLUTChannel (unsigned int *bank*, unsigned int *channel*, unsigned int *sizeEntries*, array< unsigned int >^ *entries*) [virtual]

Get the LUT channel settings from the camera.

Parameters

<i>bank</i>	Bank to retrieve.
<i>channel</i>	Channel to retrieve.
<i>sizeEntries</i>	Number of entries in LUT table to read.
<i>entries</i>	Array to store LUT entries in.

See also

[GetLUTInfo\(\)](#)
[EnableLUT\(\)](#)
[SetLUTChannel\(\)](#)

7.29.3.13 LutData GetLUTInfo () [virtual]

Query if LUT support is available on the camera.

See also

[EnableLUT\(\)](#)
[GetLUTChannel\(\)](#)
[SetLUTChannel\(\)](#)

Returns

[LutData](#) structure containing the LUT information.

7.29.3.14 unsigned int GetMemoryChannel () [virtual]

Retrieve the current memory channel from the camera.

See also

[SaveToMemoryChannel\(\)](#)
[RestoreFromMemoryChannel\(\)](#)
[GetMemoryChannelInfo\(\)](#)

Returns

Currently selected memory channel.

7.29.3.15 unsigned int GetMemoryChannelInfo () [virtual]

Query the camera for memory channel support.

If the number of channels is 0, then memory channel support is not available.

See also

[GetMemoryChannel\(\)](#)
[SaveToMemoryChannel\(\)](#)
[RestoreFromMemoryChannel\(\)](#)

Returns

Number of memory channels supported.

7.29.3.16 FlyCapture2::CameraBase * GetNativeCamera () [package]

7.29.3.17 CameraProperty GetProperty (PropertyType type) [virtual]

Reads the settings for the specified property from the camera.

If auto is on, the integer and abs values returned may not be consistent with each other.

Parameters

<i>type</i>	The PropertyType to retrieve information about.
-------------	---

See also

[GetPropertyInfo\(\)](#)
[SetProperty\(\)](#)

Returns

Property structure containing property information.

7.29.3.18 CameraPropertyInfo GetPropertyInfo (PropertyType *type*) [virtual]

Retrieves information about the specified camera property.

Parameters

<i>type</i>	The PropertyType to retrieve information about.
-------------	---

See also

[GetProperty\(\)](#)

[SetProperty\(\)](#)

Returns

PropertyInfo structure containing property information.

7.29.3.19 System::String GetRegisterString (unsigned int *registerVal*) [static]

Returns a text representation of the register value.

Parameters

<i>registerVal</i>	The register value to query.
--------------------	------------------------------

Returns

The text representation of the register.

7.29.3.20 StrobeControl GetStrobe (unsigned int *source*) [virtual]

Retrieve current strobe settings from the camera.

Parameters

<i>source</i>	Source pin for strobe information.
---------------	------------------------------------

See also

[GetStrobeInfo\(\)](#)

[SetStrobe\(\)](#)

Returns

[StrobeControl](#) structure containing strobe information.

7.29.3.21 StrobeInfo GetStrobeInfo (unsigned int *source*) [virtual]

Retrieve strobe information from the camera.

Parameters

<i>source</i>	Source pin for strobe information.
---------------	------------------------------------

See also

[GetStrobe\(\)](#)

[SetStrobe\(\)](#)

Returns

[StrobeInfo](#) structure containing strobe information.

7.29.3.22 TriggerMode GetTriggerMode () [virtual]

Retrieve current trigger settings from the camera.

See also

[GetTriggerModeInfo\(\)](#)

[SetTriggerMode\(\)](#)

Returns

[TriggerMode](#) structure containing trigger mode settings.

7.29.3.23 TriggerModeInfo GetTriggerModeInfo () [virtual]

Retrieve trigger information from the camera.

See also

[GetTriggerMode\(\)](#)

[SetTriggerMode\(\)](#)

Returns

[TriggerModeInfo](#) structure containing receive trigger information.

7.29.3.24 bool IsConnected () [virtual]

Checks if the [ManagedCamera](#) object is currently connected to a physical camera.

See also

[Connect\(\)](#)
[Disconnect\(\)](#)

Returns

Whether the [ManagedCamera](#) object is connected to a physical camera.

7.29.3.25 void OnNativeCallback (FlyCapture2::Image * *pImage*, void * *pCallbackData*) [protected]

7.29.3.26 unsigned int ReadRegister (unsigned int *address*) [virtual]

Read the specified register from the camera.

Parameters

<i>address</i>	DCAM address to be read from.
----------------	-------------------------------

See also

[WriteRegister\(\)](#)

Returns

The register value that is read.

7.29.3.27 void ReadRegisterBlock (unsigned short *addressHigh*, unsigned int *addressLow*, array< unsigned int >^ *buffer*) [virtual]

Read from the specified register block on the camera.

Parameters

<i>addressHigh</i>	Top 16 bits of the 48 bit absolute address to read from.
<i>addressLow</i>	Bottom 32 bits of the 48 bits absolute address to read from.
<i>buffer</i>	Array to store read data.

See also

[WriteRegisterBlock\(\)](#)

7.29.3.28 void RestoreFromMemoryChannel (unsigned int *channel*) [virtual]

Restore the specified current memory channel.

Parameters

<i>channel</i>	Memory channel to restore from.
----------------	---------------------------------

See also

[GetMemoryChannel\(\)](#)
[SaveToMemoryChannel\(\)](#)
[GetMemoryChannelInfo\(\)](#)

7.29.3.29 void RetrieveBuffer (ManagedImage^ *image*) [virtual]

Retrieves the the next image object containing the next image.

If the grab mode has not been set, or has been set to DROP_FRAMES the default behavior is to requeue images for DMA if they have not been retrieved by the time the next image transfer completes. If BUFFER_FRAMES is specified, the next image in the sequence will be retrieved. Note that for the BUFFER_FRAMES case, if retrieval does not keep up with the DMA process, images will be lost. The default behavior is to perform DROP_FRAMES image retrieval.

Parameters

<i>image</i>	ManagedImage object to store image data.
--------------	--

See also

[StartCapture\(\)](#)
[StopCapture\(\)](#)
[WaitForBufferEvent\(\)](#)

7.29.3.30 void SaveToMemoryChannel (unsigned int *channel*) [virtual]

Save the current settings to the specified current memory channel.

Parameters

<i>channel</i>	Memory channel to save to.
----------------	----------------------------

See also

[GetMemoryChannel\(\)](#)
[RestoreFromMemoryChannel\(\)](#)
[GetMemoryChannelInfo\(\)](#)

7.29.3.31 void SetActiveLUTBank (unsigned int *activeBank*) [virtual]

Set the LUT bank that will be used.

Parameters

<i>activeBank</i>	The bank to be set as active.
-------------------	-------------------------------

7.29.3.32 void SetCallback (ImageEventCallback^ *hCallbackDelegate*) [virtual]

Sets the callback data to be used on completion of image transfer.

To clear the current stored callback data, pass in NULL as the argument.

Parameters

<i>hCallback-Delegate</i>	A function to be called when a new image is received.
---------------------------	---

See also

[StartCapture\(\)](#)

Returns

An Error indicating the success or failure of the function.

7.29.3.33 void SetCamera (System::IntPtr *otherCamera*)

Set camera from a integer pointer camera.

7.29.3.34 void SetConfiguration (FC2Config^ *config*) [virtual]

Set the configuration associated with the camera object.

Parameters

<i>config</i>	Configuration structure to be used.
---------------	-------------------------------------

See also

[GetConfiguration\(\)](#)

7.29.3.35 void SetEmbeddedImageInfo (EmbeddedImageInfo^ *info*) [virtual]

Sets the on/off values of the embedded image information structure to the camera.

Parameters

<i>info</i>	Structure to be used.
-------------	-----------------------

See also

[GetEmbeddedImageInfo\(\)](#)

7.29.3.36 void SetGPIOPinDirection (unsigned int *pin*, unsigned int *direction*)
[virtual]

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters

<i>pin</i>	Pin to get the direction for.
<i>direction</i>	Direction of the pin. 0 for input, 1 for output.

See also

[GetGPIOPinDirection\(\)](#)

7.29.3.37 void SetGPIOPinDirection (unsigned int *pin*, unsigned int *direction*, bool *broadcast*)
[virtual]

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters

<i>pin</i>	Pin to get the direction for.
<i>direction</i>	Direction of the pin. 0 for input, 1 for output.
<i>broadcast</i>	Whether the action should be broadcast.

See also

[GetGPIOPinDirection\(\)](#)

7.29.3.38 void SetLUTChannel (unsigned int *bank*, unsigned int *channel*, unsigned int *sizeEntries*, array< unsigned int >^ *entries*) [virtual]

Set the LUT channel settings to the camera.

Parameters

<i>bank</i>	Bank to set.
<i>channel</i>	Channel to set.
<i>sizeEntries</i>	Number of entries in LUT table to write. This must be the same size as numEntries returned by GetLutInfo().
<i>entries</i>	Array containing LUT entries to write.

See also

[GetLUTInfo\(\)](#)

[EnableLUT\(\)](#)

[GetLUTChannel\(\)](#)

7.29.3.39 void SetProperty (CameraProperty^ *camProperty*) [virtual]

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

Parameters

<i>camProperty</i>	CameraProperty structure to be used.
--------------------	--

See also

[GetPropertyInfo\(\)](#)

[GetProperty\(\)](#)

7.29.3.40 void SetProperty (CameraProperty^ *camProperty*, bool *broadcast*) [virtual]

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

Parameters

<i>camProperty</i>	CameraProperty structure to be used.
<i>broadcast</i>	Whether the action should be broadcast.

See also

[GetPropertyInfo\(\)](#)[GetProperty\(\)](#)**7.29.3.41 void SetStrobe ([StrobeControl](#)^ *strobeControl*) [virtual]**

Set current strobe settings to the camera.

The strobe pin must be specified in the structure before being passed in to the function.

Parameters

<i>strobe- Control</i>	Structure providing strobe settings.
----------------------------	--------------------------------------

See also

[GetStrobeInfo\(\)](#)[GetStrobe\(\)](#)**7.29.3.42 void SetTriggerMode ([TriggerMode](#)^ *triggerMode*) [virtual]**

Set the specified trigger settings to the camera.

Parameters

<i>triggerMode</i>	Structure providing trigger mode settings.
--------------------	--

See also

[GetTriggerModeInfo\(\)](#)[GetTriggerMode\(\)](#)**7.29.3.43 void StartCapture () [virtual]**

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera. [RetrieveBuffer\(\)](#) can be called to get the image data.

See also

[RetrieveBuffer\(\)](#)
[StopCapture\(\)](#)

7.29.3.44 void StartCapture (ImageEventCallback^ *hCallbackDelegate*) [virtual]

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera. The callback function parameter is called on completion of image transfer.

Parameters

<i>hCallback-Delegate</i>	A function to be called when a new image is received.
---------------------------	---

See also

[RetrieveBuffer\(\)](#)
[StopCapture\(\)](#)

7.29.3.45 void StopCapture () [virtual]

Stops isochronous image transfer and cleans up all associated resources.

See also

[StartCapture\(\)](#)
[RetrieveBuffer\(\)](#)

7.29.3.46 void WaitForBufferEvent (ManagedImage^ *image*, unsigned int *eventNumber*) [virtual]

Retrieves the next image event containing the next part of the image.

Parameters

<i>image</i>	ManagedImage object to store image data.
<i>event-Number</i>	The event number to wait for.

See also

[RetrieveBuffer](#)

7.29.3.47 `void WriteRegister (unsigned int address, unsigned int value)` [virtual]

Write to the specified register on the camera.

Parameters

<i>address</i>	DCAM address to be written to.
<i>value</i>	The value to be written.

See also

[ReadRegister\(\)](#)

7.29.3.48 `void WriteRegister (unsigned int address, unsigned int value, bool broadcast)`
[virtual]

Write to the specified register on the camera.

Parameters

<i>address</i>	DCAM address to be written to.
<i>value</i>	The value to be written.
<i>broadcast</i>	Whether the action should be broadcast.

See also

[ReadRegister\(\)](#)

7.29.3.49 `void WriteRegisterBlock (unsigned short addressHigh, unsigned int addressLow,
array< unsigned int >^ buffer)` [virtual]

Write to the specified register block on the camera.

Parameters

<i>addressHigh</i>	Top 16 bits of the 48 bit absolute address to write to.
<i>addressLow</i>	Bottom 32 bits of the 48 bits absolute address to write to.
<i>buffer</i>	Array containing data to be written.

See also

[ReadRegisterBlock\(\)](#)

7.29.4 Member Data Documentation

7.29.4.1 `ImageEventCallback ^ m_externalDelegate` [protected]

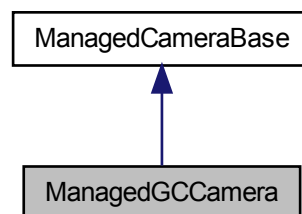
7.29.4.2 `ImageCallbackDelegate ^ m_internalDelegate` [protected]

7.29.4.3 `bool m_isLocal` [protected]

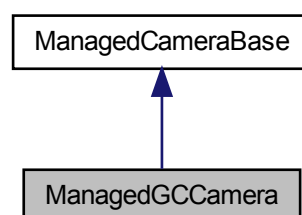
7.29.4.4 `FlyCapture2::CameraBase* m_pNativeCamBase` [protected]

7.30 ManagedGCCamera Class Reference

Inheritance diagram for ManagedGCCamera:



Collaboration diagram for ManagedGCCamera:



Public Member Functions

- [ManagedGCCamera](#) (void)
- virtual [~ManagedGCCamera](#) (void)

- virtual void [Connect](#) ([ManagedPGRGuid](#)[^] mgdPGRGuid) override
Connects the [ManagedCamera](#) object to the camera specified by the GUID.
- virtual void [Connect](#) ([ManagedPGRGuid](#)[^] mgdPGRGuid, String[^] xmlPath) override
- virtual void [Disconnect](#) (void) override
Disconnects the [ManagedCamera](#) object from the camera.
- void [SetCamera](#) ([ManagedCameraBase](#)[^] cameraBase)
- void [SetCamera](#) ([ManagedCameraBase](#)[^] cameraBase, String[^] xmlPath)
- GenICam::GenApi::NodeMap[^] [GetNodeMap](#) ()

Protected Member Functions

- [!ManagedGCCamera](#) ()

7.30.1 Constructor & Destructor Documentation

7.30.1.1 [ManagedGCCamera](#) (void)

7.30.1.2 [~ManagedGCCamera](#) (void) [virtual]

7.30.1.3 [!ManagedGCCamera](#) () [protected]

7.30.2 Member Function Documentation

7.30.2.1 void [Connect](#) ([ManagedPGRGuid](#)[^] *mgdPGRGuid*) [override, virtual]

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

7.30.2.2 void [Connect](#) ([ManagedPGRGuid](#)[^] *mgdPGRGuid*, String[^] *xmlPath*) [override, virtual]

7.30.2.3 void [Disconnect](#) (void) [override, virtual]

Disconnects the [ManagedCamera](#) object from the camera.

This allows another physical camera to be connected to the [ManagedCamera](#) object.

See also

[Connect\(\)](#)

Reimplemented from [ManagedCameraBase](#).

7.30.2.4 GenICam::GenApi::NodeMap GetNodeMap ()

7.30.2.5 void SetCamera (ManagedCameraBase^ cameraBase)

7.30.2.6 void SetCamera (ManagedCameraBase^ cameraBase, String^ xmlPath)

7.31 ManagedGCPort Class Reference

Public Member Functions

- [ManagedGCPort](#) (GCCamera *camera)
- virtual [~ManagedGCPort](#) (void)
- virtual void [Read](#) (IntPtr buffer, __int64 address, __int64 length) override
- virtual void [Write](#) (IntPtr buffer, __int64 address, __int64 length) override

7.31.1 Constructor & Destructor Documentation

7.31.1.1 ManagedGCPort (GCCamera * camera)

7.31.1.2 ~ManagedGCPort (void) [virtual]

7.31.2 Member Function Documentation

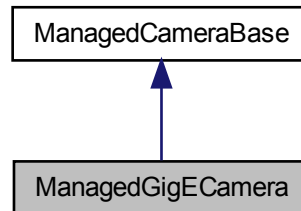
7.31.2.1 void Read (IntPtr buffer, __int64 address, __int64 length) [override, virtual]

7.31.2.2 void Write (IntPtr buffer, __int64 address, __int64 length) [override, virtual]

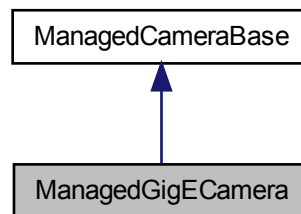
7.32 ManagedGigECamera Class Reference

The GigECamera object represents a physical Gigabit Ethernet camera.

Inheritance diagram for ManagedGigECamera:



Collaboration diagram for ManagedGigECamera:



Public Member Functions

- [ManagedGigECamera](#) ()
- [~ManagedGigECamera](#) ()
- virtual void [Connect](#) ([ManagedPGRGuid](#)[^] mgdPGRGuid) override
Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Protected Member Functions

- [!ManagedGigECamera](#) ()

GVCP Register Operation

These functions deal with GVCP register operation on the camera.

- void [WriteGVCPRegister](#) (unsigned int address, unsigned int value)
Write a GVCP register.
- void [WriteGVCPRegister](#) (unsigned int address, unsigned int value, bool broadcast)
Write a GVCP register.
- unsigned int [ReadGVCPRegister](#) (unsigned int address)
Read a GVCP register.
- void [WriteGVCPRegisterBlock](#) (unsigned int address, array< unsigned int >^buffer)
Write a GVCP register block.
- void [ReadGVCPRegisterBlock](#) (unsigned int address, array< unsigned int >^buffer)
Read a GVCP register block.
- void [WriteGVCPMemory](#) (unsigned int address, array< unsigned char >^buffer)
Write a GVCP memory block.
- void [ReadGVCPMemory](#) (unsigned int address, array< unsigned char >^buffer)
Read a GVCP memory block.

GigE property manipulation

These functions deal with GigE properties.

- [GigEProperty](#)^ [GetGigEProperty](#) (GigEPropertyType propType)
Get the specified GigEProperty.
- void [SetGigEProperty](#) (GigEProperty^ prop)
Set the specified GigEProperty.
- unsigned int [DiscoverGigEPacketSize](#) ()
Discover the largest packet size that works for the network link between the PC and the camera.

GigE image settings

These functions deal with GigE image setting.

- bool [QueryGigEImagingMode](#) (Mode mode)
Check if the particular imaging mode is supported by the camera.
- Mode [GetGigEImagingMode](#) ()
Get the current imaging mode on the camera.

- void [SetGigEImagingMode](#) ([Mode](#) mode)
Set the current imaging mode to the camera.
- [GigEImageSettingsInfo](#)^ [GetGigEImageSettingsInfo](#) ()
Get information about the image settings possible on the camera.
- [GigEImageSettings](#)^ [GetGigEImageSettings](#) ()
Get the current image settings on the camera.
- void [SetGigEImageSettings](#) ([GigEImageSettings](#)^ settings)
Set the image settings specified to the camera.

GigE Configuration

These functions deal with configuring camera.

- [GigEConfig](#)^ [GetGigEConfig](#) ()
Get the current configuration on the camera.
- void [SetGigEConfig](#) ([GigEConfig](#)^ config)
Set the configuration specified to the camera.

GigE image binning settings

These functions deal with GigE image binning setting.

- void [GetGigEImageBinningSettings](#) (unsigned int% horzBinningValue, unsigned int% vertBinningValue)
Get the current binning settings on the camera.
- void [SetGigEImageBinningSettings](#) (unsigned int horzBinningValue, unsigned int vertBinningValue)
Set the specified binning values to the camera.

GigE image stream configuration

These functions deal with GigE image stream configuration.

- unsigned int [GetNumStreamChannels](#) ()
Get the number of stream channels present on the camera.
- [GigEStreamChannel](#)^ [GetGigEStreamChannelInfo](#) (unsigned int channel)
Get the stream channel information for the specified channel.
- void [SetGigEStreamChannelInfo](#) (unsigned int channel, [GigEStreamChannel](#)^ channelInfo)
Set the stream channel information for the specified channel.

7.32.1 Detailed Description

The GigECamera object represents a physical Gigabit Ethernet camera.

The object must first be connected to using [Connect\(\)](#) before any other operations can proceed.

Please see [ManagedCameraBase](#) for basic functions that this class inherits from.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 ManagedGigECamera ()

7.32.2.2 ~ManagedGigECamera ()

7.32.2.3 !ManagedGigECamera () [protected]

7.32.3 Member Function Documentation

7.32.3.1 void Connect (ManagedPGRGuid^ *mgdPGRGuid*) [override, virtual]

Connects the [ManagedCamera](#) object to the camera specified by the GUID.

Parameters

<i>mgdPGR-Guid</i>	The unique identifier for a specific camera on the PC.
--------------------	--

See also

[ManagedBusManager::GetCameraFromIndex\(\)](#)
[ManagedBusManager::GetCameraFromSerialNumber\(\)](#)

Reimplemented from [ManagedCameraBase](#).

7.32.3.2 unsigned int DiscoverGigEPacketSize ()

Discover the largest packet size that works for the network link between the PC and the camera.

This is useful in cases where there may be multiple links between the PC and the camera and there is a possibility of a component not supporting the recommended jumbo frame packet size of 9000.

Returns

The maximum packet size supported by the link.

7.32.3.3 **GigEConfig** GetGigEConfig ()

Get the current configuration on the camera.

Returns

Current configuration on camera.

7.32.3.4 **void** GetGigEImageBinningSettings (unsigned int% *horzBinningValue*, unsigned int% *vertBinningValue*)

Get the current binning settings on the camera.

Parameters

<i>horzBinningValue</i>	Current horizontal binning value.
<i>vertBinningValue</i>	Current vertical binning value.

7.32.3.5 **GigEImageSettings** GetGigEImageSettings ()

Get the current image settings on the camera.

Returns

Current image settings on camera.

7.32.3.6 **GigEImageSettingsInfo** GetGigEImageSettingsInfo ()

Get information about the image settings possible on the camera.

Returns

Image settings information.

7.32.3.7 **Mode** GetGigEImagingMode ()

Get the current imaging mode on the camera.

Returns

Current imaging mode on the camera.

7.32.3.8 GigEProperty GetGigEProperty (GigEPropertyType *propType*)

Get the specified [GigEProperty](#).

Returns

The GigE property to get.

7.32.3.9 GigEStreamChannel GetGigEStreamChannelInfo (unsigned int *channel*)

Get the stream channel information for the specified channel.

Parameters

<i>channel</i>	Channel number to use.
----------------	------------------------

Returns

Stream channel information for the specified channel.

7.32.3.10 unsigned int GetNumStreamChannels ()

Get the number of stream channels present on the camera.

Returns

Number of stream channels present.

7.32.3.11 bool QueryGigEImagingMode (Mode *mode*)

Check if the particular imaging mode is supported by the camera.

Parameters

<i>mode</i>	The mode to check.
-------------	--------------------

Returns

Whether the mode is supported.

7.32.3.12 void ReadGVCPMemory (unsigned int *address*, array< unsigned char >^ *buffer*)

Read a GVCP memory block.

Parameters

<i>address</i>	GVCP address to be read from.
<i>buffer</i>	Array for data to be read into.

7.32.3.13 unsigned int ReadGVCPRegister (unsigned int *address*)

Read a GVCP register.

Parameters

<i>address</i>	GVCP address to be read from.
----------------	-------------------------------

Returns

The value that is read.

7.32.3.14 void ReadGVCPRegisterBlock (unsigned int *address*, array< unsigned int >^ *buffer*)

Read a GVCP register block.

Parameters

<i>address</i>	GVCP address to be read from.
<i>buffer</i>	Array for data to be read into.

7.32.3.15 void SetGigEConfig (GigEConfig^ *config*)

Set the configuration specified to the camera.

Parameters

<i>config</i>	Configuration to set to camera.
---------------	---------------------------------

7.32.3.16 void SetGigEImageBinningSettings (unsigned int *horzBinningValue*, unsigned int *vertBinningValue*)

Set the specified binning values to the camera.

It is recommended that [GetGigEImageSettingsInfo\(\)](#) be called after this function succeeds to retrieve the new image settings information for the new binning mode.

Parameters

<i>horz-Binning-Value</i>	Horizontal binning value.
<i>vert-Binning-Value</i>	Vertical binning value.

7.32.3.17 void SetGigEImageSettings (GigEImageSettings^ settings)

Set the image settings specified to the camera.

Parameters

<i>settings</i>	Image settings to set to camera.
-----------------	----------------------------------

7.32.3.18 void SetGigEImagingMode (Mode mode)

Set the current imaging mode to the camera.

This should only be done when the camera is not streaming images.

Parameters

<i>mode</i>	Imaging mode to set to the camera.
-------------	------------------------------------

7.32.3.19 void SetGigEProperty (GigEProperty^ prop)

Set the specified [GigEProperty](#).

The GigEPropertyType field must be set in order for this function to succeed.

Parameters

<i>prop</i>	The GigE property to set.
-------------	---------------------------

7.32.3.20 void SetGigEStreamChannelInfo (unsigned int channel, GigEStreamChannel^ channelInfo)

Set the stream channel information for the specified channel.

Parameters

<i>channel</i>	Channel number to use.
<i>channelInfo</i>	Stream channel information to use for the specified channel.

7.32.3.21 void WriteGVCPMemory (unsigned int *address*, array< unsigned char >^ *buffer*)

Write a GVCP memory block.

Parameters

<i>address</i>	GVCP address to be write to.
<i>buffer</i>	Array containing data to be written.

7.32.3.22 void WriteGVCPRegister (unsigned int *address*, unsigned int *value*)

Write a GVCP register.

Parameters

<i>address</i>	GVCP address to be written to.
<i>value</i>	The value to be written.

7.32.3.23 void WriteGVCPRegister (unsigned int *address*, unsigned int *value*, bool *broadcast*)

Write a GVCP register.

Parameters

<i>address</i>	GVCP address to be written to.
<i>value</i>	The value to be written.
<i>broadcast</i>	Whether the action should be broadcast.

7.32.3.24 void WriteGVCPRegisterBlock (unsigned int *address*, array< unsigned int >^ *buffer*)

Write a GVCP register block.

Parameters

<i>address</i>	GVCP address to be write to.
<i>buffer</i>	Array containing data to be written.

7.33 ManagedImage Class Reference

The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.

Public Member Functions

- [ManagedImage](#) ()
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char *pData, unsigned int dataSize, [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char *pData, unsigned int dataSize, unsigned int receivedActualSize, - [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char *pData, unsigned int dataSize, [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, unsigned int stride, unsigned char *pData, unsigned int dataSize, unsigned int receivedActualSize, - [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) (unsigned char *pData, unsigned int dataSize)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, [PixelFormat](#) format)
- [ManagedImage](#) (unsigned int rows, unsigned int cols, [PixelFormat](#) format, [BayerTileFormat](#) bayerFormat)
- [ManagedImage](#) ([ManagedImage](#)^ image)
- [~ManagedImage](#) ()
- void [SetDimensions](#) (unsigned int rows, unsigned int cols, unsigned int stride, [PixelFormat](#) pixelFormat, [BayerTileFormat](#) bayerFormat)

Sets the dimensions of the [ManagedImage](#) object.
- void [SetData](#) (unsigned char *pData, unsigned int dataSize)

Set the data of the [ManagedImage](#) object.
- void [CalculateStatistics](#) ([ManagedImageStatistics](#)^ statistics)

Calculate statistics associated with the image.
- void [Save](#) (System::String^ fileName)

Save the image to the specified file name.
- void [Save](#) (System::String^ fileName, [ImageFileFormat](#) format)

Save the image to the specified file name with the file format specified.
- void [Save](#) (System::String^ fileName, [PngOption](#)^ option)

Save the image to the specified file name with the options specified.
- void [Save](#) (System::String^ fileName, [PpmOption](#)^ option)

Save the image to the specified file name with the options specified.
- void [Save](#) (System::String^ fileName, [PgmOption](#)^ option)

Save the image to the specified file name with the options specified.
- void [Save](#) (System::String^ fileName, [TiffOption](#)^ option)

Save the image to the specified file name with the options specified.
- void [Save](#) (System::String^ fileName, [JpegOption](#)^ option)

Save the image to the specified file name with the options specified.
- void [Save](#) (System::String^ fileName, [Jpg2Option](#)^ option)

Save the image to the specified file name with the options specified.
- void [Convert](#) ([ManagedImage](#)^ destImage)

Converts the current image buffer and stores the result in the specified image.

- void [Convert](#) ([PixelFormat](#) format, [ManagedImage](#)[^] destImage)
Converts the current image buffer to the specified output format and stores the result in the specified image.
- void [ReleaseBuffer](#) ()
Release the buffer associated with the [ManagedImage](#).
- void * [GetRawNativeImagePointer](#) ()

Static Public Member Functions

- static unsigned int [DetermineBitsPerPixel](#) ([PixelFormat](#) format)
Calculate the bits per pixel for the specified pixel format.

Protected Member Functions

- [!ManagedImage](#) ()

Package Functions

- [ManagedImage](#) (FlyCapture2::Image &image)
- bool [IsNativeImageValid](#) ()
- FlyCapture2::Image * [GetNativeImage](#) ()

Properties

- static [ColorProcessingAlgorithm](#) [defaultColorProcessingAlgorithm](#) [get, set]
The default color processing algorithm to be used.
- static [PixelFormat](#) [defaultOutputPixelFormat](#) [get, set]
The default output pixel format to be used.
- [ColorProcessingAlgorithm](#) [colorProcessingAlgorithm](#) [get, set]
Color processing algorithm to be used.
- [PixelFormat](#) [pixelFormat](#) [get]
Pixel format of the image.
- [BayerTileFormat](#) [bayerTileFormat](#) [get]
Bayer tile format of the image.
- unsigned int [cols](#) [get]
Number of columns in the image.
- unsigned int [rows](#) [get]
Number of rows in the image.
- unsigned int [stride](#) [get]
Number of bytes between rows in the image.
- unsigned int [bitsPerPixel](#) [get]
Number of bits per pixel in the image.

- unsigned char * [data](#) [get]
Raw pointer to image data.
- unsigned int [receivedDataSize](#) [get]
Actual received data size.
- [ImageMetadata](#)^ [imageMetadata](#) [get]
Get the metadata associated with the image.
- [TimeStamp](#)^ [timeStamp](#) [get]
Get the timestamp data associated with the image.
- [System::Drawing::Bitmap](#)^ [bitmap](#) [get]
Get the internal bitmap representation associated with the image.

7.33.1 Detailed Description

The ManagedImage class is used to retrieve images from a camera, convert between multiple pixel formats and save images to disk.

Operations on Image objects are not guaranteed to be thread safe. It is recommended that operations on Image objects be protected by thread synchronization constructs such as mutexes.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 ManagedImage ()

7.33.2.2 ManagedImage (unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char * *pData*, unsigned int *dataSize*, PixelFormat *format*)

7.33.2.3 ManagedImage (unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char * *pData*, unsigned int *dataSize*, unsigned int *receivedActualSize*, PixelFormat *format*)

7.33.2.4 ManagedImage (unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char * *pData*, unsigned int *dataSize*, PixelFormat *format*, BayerTileFormat *bayerFormat*)

7.33.2.5 ManagedImage (unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, unsigned char * *pData*, unsigned int *dataSize*, unsigned int *receivedActualSize*, PixelFormat *format*, BayerTileFormat *bayerFormat*)

7.33.2.6 ManagedImage (unsigned char * *pData*, unsigned int *dataSize*)

7.33.2.7 ManagedImage (unsigned int *rows*, unsigned int *cols*, PixelFormat *format*)

7.33.2.8 ManagedImage (unsigned int *rows*, unsigned int *cols*, PixelFormat *format*, BayerTileFormat *bayerFormat*)

7.33.2.9 **ManagedImage** (**ManagedImage**[^] *image*)

7.33.2.10 **~ManagedImage** ()

7.33.2.11 **ManagedImage** (**FlyCapture2::Image** & *image*) [package]

7.33.2.12 **!ManagedImage** () [protected]

7.33.3 Member Function Documentation

7.33.3.1 **void CalculateStatistics** (**ManagedImageStatistics**[^] *statistics*)

Calculate statistics associated with the image.

In order to collect statistics for a particular channel, the enabled flag for the channel must be set to true. Statistics can only be collected for images in Mono8, Mono16, RGB, RGBU, BGR and BGRU.

Parameters

<i>statistics</i>	The ManagedImageStatistics object to hold the statistics.
-------------------	---

7.33.3.2 **void Convert** (**ManagedImage**[^] *destImage*)

Converts the current image buffer 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.

Parameters

<i>destImage</i>	Destination image.
------------------	--------------------

7.33.3.3 **void Convert** (**PixelFormat** *format*, **ManagedImage**[^] *destImage*)

Converts the current image buffer to the specified output 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.

Parameters

<i>format</i>	Output format of the converted image.
<i>destImage</i>	Destination image.

7.33.3.4 unsigned int DetermineBitsPerPixel (PixelFormat *format*) [static]

Calculate the bits per pixel for the specified pixel format.

Parameters

<i>format</i>	The pixel format.
---------------	-------------------

Returns

The bits per pixel.

7.33.3.5 FlyCapture2::Image * GetNativeImage () [package]**7.33.3.6** void * GetRawNativeImagePointer ()**7.33.3.7** bool IsNativeImageValid () [package]**7.33.3.8** void ReleaseBuffer ()

Release the buffer associated with the [ManagedImage](#).

If no buffer is associated, the function does nothing.

7.33.3.9 void Save (System::String^ *fileName*)

Save the image to the specified file name.

Parameters

<i>fileName</i>	Filename to save image with.
-----------------	------------------------------

7.33.3.10 void Save (System::String^ *fileName*, ImageFileFormat *format*)

Save the image to the specified file name with the file format specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>format</i>	File format to save in.

7.33.3.11 void Save (System::String^ *fileName*, PngOption^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.12 void Save (System::String^ *fileName*, PpmOption^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.13 void Save (System::String^ *fileName*, PgmOption^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.14 void Save (System::String^ *fileName*, TiffOption^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.15 void Save (System::String^ *fileName*, JpegOption^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.16 void Save (System::String^ *fileName*, Jpg2Option^ *option*)

Save the image to the specified file name with the options specified.

Parameters

<i>fileName</i>	Filename to save image with.
<i>option</i>	Options to use while saving image.

7.33.3.17 void SetData (unsigned char * *pData*, unsigned int *dataSize*)

Set the data of the [ManagedImage](#) object.

Ownership of the image buffer is not transferred to the [ManagedImage](#) object. It is the user's responsibility to delete the buffer when it is no longer in use.

Parameters

<i>pData</i>	Pointer to the image buffer.
<i>dataSize</i>	Size of the image buffer.

7.33.3.18 void SetDimensions (unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, [PixelFormat](#) *pixelFormat*, [BayerTileFormat](#) *bayerFormat*)

Sets the dimensions of the [ManagedImage](#) object.

Parameters

<i>rows</i>	Number of rows to set.
<i>cols</i>	Number of cols to set.
<i>stride</i>	Stride to set.
<i>pixelFormat</i>	Pixel format to set.
<i>bayerFormat</i>	Bayer tile format to set.

7.33.4 Property Documentation

7.33.4.1 [BayerTileFormat](#) *bayerTileFormat* [get]

Bayer tile format of the image.

7.33.4.2 [System::Drawing::Bitmap](#)[^] *bitmap* [get]

Get the internal bitmap representation associated with the image.

Returns

A [System::Drawing::Bitmap](#) containing the image data.

7.33.4.3 unsigned int bitsPerPixel [get]

Number of bits per pixel in the image.

7.33.4.4 ColorProcessingAlgorithm colorProcessingAlgorithm [get, set]

Color processing algorithm to be used.

7.33.4.5 unsigned int cols [get]

Number of columns in the image.

7.33.4.6 unsigned char* data [get]

Raw pointer to image data.

7.33.4.7 ColorProcessingAlgorithm defaultColorProcessingAlgorithm [static, get, set]

The default color processing algorithm to be used.

7.33.4.8 PixelFormat defaultOutputPixelFormat [static, get, set]

The default output pixel format to be used.

7.33.4.9 ImageMetadata^ imageMetadata [get]

Get the metadata associated with the image.

This includes embedded image information.

Returns

Metadata associated with the image.

7.33.4.10 PixelFormat pixelFormat [get]

Pixel format of the image.

7.33.4.11 unsigned int receivedDataSize [get]

Actual received data size.

7.33.4.12 unsigned int rows [get]

Number of rows in the image.

7.33.4.13 unsigned int stride [get]

Number of bytes between rows in the image.

7.33.4.14 TimeStamp[^] timeStamp [get]

Get the timestamp data associated with the image.

Returns

Timestamp data associated with the image.

7.34 ManagedImageStatistics Class Reference

Public Member Functions

- [ManagedImageStatistics](#) ()
- [~ManagedImageStatistics](#) ()
- void [EnableAll](#) ()
- void [DisableAll](#) ()
- void [EnableGreyOnly](#) ()
- void [EnableRGBOnly](#) ()
- void [EnableHSLOnly](#) ()
- bool [GetChannelStatus](#) ([StatisticsChannel](#) channel)
- void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)
- void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int% min, unsigned int% max)
- void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int% pixelValueMin, unsigned int% pixelValueMax)
- void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int% numPixelValues)
- void [GetMean](#) ([StatisticsChannel](#) channel, float% mean)
- void [GetHistogram](#) ([StatisticsChannel](#) channel, array< int >^histogram)
- void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int% rangeMin, unsigned int% rangeMax, unsigned int% pixelValueMin, unsigned int% pixelValueMax, unsigned int% numPixelValues, float% mean, array< int >^histogram)

Package Functions

- [FlyCapture2::ImageStatistics * GetNativeImageStatistics](#) ()

7.34.1 Constructor & Destructor Documentation

7.34.1.1 `ManagedImageStatistics ()`

7.34.1.2 `~ManagedImageStatistics ()`

7.34.2 Member Function Documentation

7.34.2.1 `void DisableAll ()`

7.34.2.2 `void EnableAll ()`

7.34.2.3 `void EnableGreyOnly ()`

7.34.2.4 `void EnableHSLOnly ()`

7.34.2.5 `void EnableRGBOnly ()`

7.34.2.6 `bool GetChannelStatus (StatisticsChannel channel)`

7.34.2.7 `void GetHistogram (StatisticsChannel channel, array< int >^ histogram)`

7.34.2.8 `void GetMean (StatisticsChannel channel, float% mean)`

7.34.2.9 `FlyCapture2::ImageStatistics * GetNativeImageStatistics ()` [package]

7.34.2.10 `void GetNumPixelValues (StatisticsChannel channel, unsigned int% numPixelValues)`

7.34.2.11 `void GetPixelValueRange (StatisticsChannel channel, unsigned int% pixelValueMin, unsigned int% pixelValueMax)`

7.34.2.12 `void GetRange (StatisticsChannel channel, unsigned int% min, unsigned int% max)`

7.34.2.13 `void GetStatistics (StatisticsChannel channel, unsigned int% rangeMin, unsigned int% rangeMax, unsigned int% pixelValueMin, unsigned int% pixelValueMax, unsigned int% numPixelValues, float% mean, array< int >^ histogram)`

7.34.2.14 `void SetChannelStatus (StatisticsChannel channel, bool enabled)`

7.35 ManagedPGRGuid Class Reference

Managed version of a PGRGuid.

Public Member Functions

- [ManagedPGRGuid](#) ()
Constructor.
- [ManagedPGRGuid](#) ([ManagedPGRGuid](#)^ managedGuid)
Copy constructor.
- [ManagedPGRGuid](#) ([ManagedPGRGuid](#)% managedGuid)
Copy constructor.
- [ManagedPGRGuid](#)% [operator=](#) ([ManagedPGRGuid](#)% managedGuid)
Assignment operator.
- virtual bool [Equals](#) (Object^ obj) override
- virtual int [GetHashCode](#) () override

Static Public Member Functions

- static bool [operator==](#) ([ManagedPGRGuid](#)% left, [ManagedPGRGuid](#)% right)
Equality operator.
- static bool [operator!=](#) ([ManagedPGRGuid](#)% left, [ManagedPGRGuid](#)% right)
Inequality operator.

Public Attributes

- unsigned int [value0](#)
- unsigned int [value1](#)
- unsigned int [value2](#)
- unsigned int [value3](#)

7.35.1 Detailed Description

Managed version of a PGRGuid.

It is used to uniquely identify a camera.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 [ManagedPGRGuid](#) () [[inline](#)]

Constructor.

7.35.2.2 [ManagedPGRGuid](#) ([ManagedPGRGuid](#)^ *managedGuid*) [[inline](#)]

Copy constructor.

7.35.2.3 ManagedPGRGuid (ManagedPGRGuid% *managedGuid*) [inline]

Copy constructor.

7.35.3 Member Function Documentation

7.35.3.1 virtual bool Equals (Object^ *obj*) [inline, override, virtual]

7.35.3.2 virtual int GetHashCode () [inline, override, virtual]

7.35.3.3 static bool operator!= (ManagedPGRGuid% *left*, ManagedPGRGuid% *right*)
[inline, static]

Inequality operator.

7.35.3.4 ManagedPGRGuid % operator= (ManagedPGRGuid% *managedGuid*)
[inline]

Assignment operator.

7.35.3.5 static bool operator== (ManagedPGRGuid% *left*, ManagedPGRGuid% *right*)
[inline, static]

Equality operator.

7.35.4 Member Data Documentation

7.35.4.1 unsigned int value0

7.35.4.2 unsigned int value1

7.35.4.3 unsigned int value2

7.35.4.4 unsigned int value3

7.36 ManagedTopologyNode Class Reference

The [ManagedTopologyNode](#) class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.

Public Types

- enum [PortType](#) { [NotConnected](#) = 1, [ConnectedToParent](#), [ConnectedToChild](#) }

Possible states of a port on a node.

- enum `NodeType` { `Computer`, `Bus`, `Camera`, `Node` }

Type of node.

Public Member Functions

- virtual `~ManagedTopologyNode` ()
- `ManagedTopologyNode` (`ManagedTopologyNode`^ other)
- `ManagedTopologyNode` (`ManagedTopologyNode`% other)
- `ManagedPGRGuid`^ `GetGuid` ()

Get the PGRGuid associated with the node.

- int `GetDeviceId` ()

Get the device ID associated with the node.

- `NodeType` `GetNodeType` ()

Get the node type associated with the node.

- `InterfaceType` `GetInterfaceType` ()

Get the interface type associated with the node.

- unsigned int `GetNumChildren` ()

Get the number of child nodes.

- `ManagedTopologyNode`^ `GetChild` (unsigned int position)

Get child node located at the specified position.

- unsigned int `GetNumPorts` ()

Get the number of ports.

- `PortType` `GetPortType` (unsigned int position)

Get type of port located at the specified position.

Package Functions

- `ManagedTopologyNode` (`FlyCapture2::TopologyNode *pNode`)

Static Package Functions

- static `ManagedTopologyNode::PortType` `TranslatePortType` (`FlyCapture2::TopologyNode::PortType` portType)
- static `FlyCapture2::TopologyNode::PortType` `TranslatePortType` (`ManagedTopologyNode::PortType` portType)
- static `ManagedTopologyNode::NodeType` `TranslateNodeType` (`FlyCapture2::TopologyNode::NodeType` portType)
- static `FlyCapture2::TopologyNode::NodeType` `TranslateNodeType` (`ManagedTopologyNode::NodeType` portType)

7.36.1 Detailed Description

The [ManagedTopologyNode](#) class contains topology information that can be used to generate a tree structure of all cameras and devices connected to a computer.

7.36.2 Member Enumeration Documentation

7.36.2.1 enum NodeType

Type of node.

Enumerator:

Computer

Bus

Camera

Node

7.36.2.2 enum PortType

Possible states of a port on a node.

Enumerator:

NotConnected

ConnectedToParent

ConnectedToChild

7.36.3 Constructor & Destructor Documentation

7.36.3.1 `virtual ~ManagedTopologyNode () [inline, virtual]`

7.36.3.2 `ManagedTopologyNode (ManagedTopologyNode^ other)`

7.36.3.3 `ManagedTopologyNode (ManagedTopologyNode% other)`

7.36.3.4 `ManagedTopologyNode (FlyCapture2::TopologyNode * pNode) [package]`

7.36.4 Member Function Documentation

7.36.4.1 `ManagedTopologyNode GetChild (unsigned int position)`

Get child node located at the specified position.

Parameters

<i>position</i>	Position of the node.
-----------------	-----------------------

Returns

[ManagedTopologyNode](#) at the specified position.

7.36.4.2 int GetDeviceId ()

Get the device ID associated with the node.

Returns

Device ID of the node.

7.36.4.3 ManagedPGRGuid GetGuid ()

Get the PGRGuid associated with the node.

Returns

PGRGuid of the node.

7.36.4.4 InterfaceType GetInterfaceType ()

Get the interface type associated with the node.

Returns

Interface type of the node.

7.36.4.5 ManagedTopologyNode::NodeType GetNodeType ()

Get the node type associated with the node.

Returns

Node type of the node.

7.36.4.6 unsigned int GetNumChildren ()

Get the number of child nodes.

Returns

Number of child nodes.

7.36.4.7 unsigned int GetNumPorts ()

Get the number of ports.

Returns

Number of ports.

7.36.4.8 ManagedTopologyNode::PortType GetPortType (unsigned int *position*)

Get type of port located at the specified position.

Parameters

<i>position</i>	Position of the port.
-----------------	-----------------------

Returns

PortType at the specified position.

7.36.4.9 ManagedTopologyNode::NodeType TranslateNodeType (FlyCapture2::TopologyNode::NodeType *portType*) [static, package]

7.36.4.10 FlyCapture2::TopologyNode::NodeType TranslateNodeType (ManagedTopologyNode::NodeType *portType*) [static, package]

7.36.4.11 ManagedTopologyNode::PortType TranslatePortType (FlyCapture2::TopologyNode::PortType *portType*) [static, package]

7.36.4.12 FlyCapture2::TopologyNode::PortType TranslatePortType (ManagedTopologyNode::PortType *portType*) [static, package]

7.37 ManagedUtilities Class Reference

Static Public Member Functions

- static void [LaunchBrowser](#) (System::String^ address)
- static void [LaunchHelp](#) (System::String^ fileName)
- static void [LaunchCommand](#) (System::String^ command)

Properties

- static [SystemInfo](#)^ [systemInfo](#) [get]
- static [FC2Version](#)^ [libraryVersion](#) [get]

7.37.1 Member Function Documentation

7.37.1.1 void LaunchBrowser (System::String^ *address*) [static]

7.37.1.2 void LaunchCommand (System::String^ *command*) [static]

7.37.1.3 void LaunchHelp (System::String^ *fileName*) [static]

7.37.2 Property Documentation

7.37.2.1 FC2Version^ libraryVersion [static, get]

7.37.2.2 SystemInfo^ systemInfo [static, get]

7.38 MJPGOption Struct Reference

Options for saving MJPEG files.

Public Member Functions

- [MJPGOption \(\)](#)

Properties

- float [frameRate](#)
Frame rate of the stream.
- int [quality](#)
Image quality (1-100)

7.38.1 Detailed Description

Options for saving MJPEG files.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 MJPGOption () [inline]

7.38.3 Property Documentation

7.38.3.1 float frameRate

Frame rate of the stream.

7.38.3.2 int quality

Image quality (1-100)

7.39 PgmOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PgmOption](#) ()

Properties

- bool [binaryFile](#)
Whether to save the PPM as a binary file.

7.39.1 Detailed Description

Options for saving PGM images.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 PgmOption () [inline]

7.39.3 Property Documentation

7.39.3.1 bool binaryFile

Whether to save the PPM as a binary file.

7.40 PngOption Struct Reference

Options for saving PNG images.

Public Member Functions

- [PngOption](#) ()

Properties

- bool [interlaced](#)
Whether to save the PNG as interlaced.
- unsigned int [compressionLevel](#)
Compression level (0-9).

7.40.1 Detailed Description

Options for saving PNG images.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 `PngOption ()` `[inline]`

7.40.3 Property Documentation

7.40.3.1 unsigned int `compressionLevel`

Compression level (0-9).

0 is no compression, 9 is best compression.

7.40.3.2 bool `interlaced`

Whether to save the PNG as interlaced.

7.41 PpmOption Struct Reference

Options for saving PPM images.

Public Member Functions

- [PpmOption \(\)](#)

Properties

- bool [binaryFile](#)
Whether to save the PPM as a binary file.

7.41.1 Detailed Description

Options for saving PPM images.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 PpmOption () [inline]

7.41.3 Property Documentation

7.41.3.1 bool binaryFile

Whether to save the PPM as a binary file.

7.42 StrobeControl Struct Reference

A camera strobe.

Properties

- unsigned int [source](#)
Source value.
- bool [onOff](#)
Flag controlling on/off.
- unsigned int [polarity](#)
Signal polarity.
- float [delay](#)
Signal delay (in ms).
- float [duration](#)
Signal duration (in ms).

7.42.1 Detailed Description

A camera strobe.

7.42.2 Property Documentation

7.42.2.1 float delay

Signal delay (in ms).

7.42.2.2 float duration

Signal duration (in ms).

7.42.2.3 bool onOff

Flag controlling on/off.

7.42.2.4 unsigned int polarity

Signal polarity.

7.42.2.5 unsigned int source

Source value.

7.43 StrobelInfo Struct Reference

A camera strobe property.

Properties

- unsigned int [source](#)
Source value.
- bool [present](#)
Presence of strobe.
- bool [readOutSupported](#)
Flag indicating if strobe value can be read out.
- bool [onOffSupported](#)
Flag indicating if on/off is supported.
- bool [polaritySupported](#)
Flag indicating if polarity is supported.
- float [minValue](#)
Minimum value.
- float [maxValue](#)
Maximum value.

7.43.1 Detailed Description

A camera strobe property.

7.43.2 Property Documentation

7.43.2.1 float maxValue

Maximum value.

7.43.2.2 float minValue

Minimum value.

7.43.2.3 bool onOffSupported

Flag indicating if on/off is supported.

7.43.2.4 bool polaritySupported

Flag indicating if polarity is supported.

7.43.2.5 bool present

Presence of strobe.

7.43.2.6 bool readOutSupported

Flag indicating if strobe value can be read out.

7.43.2.7 unsigned int source

Source value.

7.44 SystemInfo Struct Reference

Description of the system.

Properties

- [OSType osType](#)
Operating system type as described by OSType.
- [System::String^ osDescription](#)
Detailed description of the operating system.
- [ByteOrder byteOrder](#)
Byte order of the system.
- [unsigned int systemMemorySize](#)
Amount of memory available on the system.
- [System::String^ cpuDescription](#)
Detailed description of the CPU.
- [unsigned int numCpuCores](#)

Number of cores on all CPUs on the system.

- System::String^ [driverList](#)

List of drivers used.

- System::String^ [libraryList](#)

List of libraries used.

- System::String^ [gpuDescription](#)

Detailed description of the GPU.

- unsigned int [screenWidth](#)

Screen resolution width in pixels.

- unsigned int [screenHeight](#)

Screen resolution height in pixels.

7.44.1 Detailed Description

Description of the system.

7.44.2 Property Documentation

7.44.2.1 ByteOrder byteOrder

Byte order of the system.

7.44.2.2 System::String^ cpuDescription

Detailed description of the CPU.

7.44.2.3 System::String^ driverList

List of drivers used.

7.44.2.4 System::String^ gpuDescription

Detailed description of the GPU.

7.44.2.5 System::String^ libraryList

List of libraries used.

7.44.2.6 unsigned int numCpuCores

Number of cores on all CPUs on the system.

7.44.2.7 `System::String^ osDescription`

Detailed description of the operating system.

7.44.2.8 `OSType osType`

Operating system type as described by `OSType`.

7.44.2.9 `unsigned int screenHeight`

Screen resolution height in pixels.

7.44.2.10 `unsigned int screenWidth`

Screen resolution width in pixels.

7.44.2.11 `unsigned int systemMemorySize`

Amount of memory available on the system.

7.45 `TiffOption Struct Reference`

Options for saving TIFF images.

Public Types

- enum `CompressionMethod` { `None` = 1, `PackBits`, `Deflate`, `AdobeDeflate`, `CcittFax3`, `CcittFax4`, `Lzw`, `Jpeg` }

Public Member Functions

- `TiffOption` ()

Properties

- `CompressionMethod compression`
Compression method to use for encoding TIFF images.

7.45.1 Detailed Description

Options for saving TIFF images.

7.45.2 Member Enumeration Documentation

7.45.2.1 enum CompressionMethod

Enumerator:

- None** Save without any compression.
- PackBits** Save using PACKBITS compression.
- Deflate** Save using DEFLATE compression (ZLIB compression).
- AdobeDeflate** Save using ADOBE DEFLATE compression.
- CcittFax3** Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
- CcittFax4** Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
- Lzw** Save using LZW compression.
- Jpeg** Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

7.45.3 Constructor & Destructor Documentation

7.45.3.1 TiffOption () [inline]

7.45.4 Property Documentation

7.45.4.1 CompressionMethod compression

Compression method to use for encoding TIFF images.

7.46 TimeStamp Struct Reference

Timestamp information.

Properties

- long long [seconds](#)
Seconds.
- unsigned int [microSeconds](#)
Microseconds.
- unsigned int [cycleSeconds](#)
1394 cycle time seconds.
- unsigned int [cycleCount](#)
1394 cycle time count.
- unsigned int [cycleOffset](#)
1394 cycle time offset.

7.46.1 Detailed Description

Timestamp information.

7.46.2 Property Documentation

7.46.2.1 unsigned int cycleCount

1394 cycle time count.

7.46.2.2 unsigned int cycleOffset

1394 cycle time offset.

7.46.2.3 unsigned int cycleSeconds

1394 cycle time seconds.

7.46.2.4 unsigned int microSeconds

Microseconds.

7.46.2.5 long long seconds

Seconds.

7.47 Translate Class Reference

Static Package Functions

- static [ErrorType](#) [translate](#) (FlyCapture2::ErrorType errorType)
- static FlyCapture2::ErrorType [translate](#) ([ErrorType](#) errorType)
- static [GrabMode](#) [translate](#) (FlyCapture2::GrabMode grabMode)
- static FlyCapture2::GrabMode [translate](#) ([GrabMode](#) grabMode)
- static [BandwidthAllocation](#) [translate](#) (FlyCapture2::BandwidthAllocation bandwidthAllocation)
- static FlyCapture2::BandwidthAllocation [translate](#) ([BandwidthAllocation](#) bandwidthAllocation)
- static [InterfaceType](#) [translate](#) (FlyCapture2::InterfaceType interfaceType)
- static FlyCapture2::InterfaceType [translate](#) ([InterfaceType](#) interfaceType)
- static [DriverType](#) [translate](#) (FlyCapture2::DriverType driverType)
- static FlyCapture2::DriverType [translate](#) ([DriverType](#) driverType)
- static [PropertyType](#) [translate](#) (FlyCapture2::PropertyType propertyType)

- static FlyCapture2::PropertyType [translate](#) ([PropertyType](#) propertyType)
- static [FrameRate](#) [translate](#) (FlyCapture2::FrameRate frmRate)
- static FlyCapture2::FrameRate [translate](#) ([FrameRate](#) frmRate)
- static [VideoMode](#) [translate](#) (FlyCapture2::VideoMode videoMode)
- static FlyCapture2::VideoMode [translate](#) ([VideoMode](#) videoMode)
- static [PixelFormat](#) [translate](#) (FlyCapture2::PixelFormat pixelFormat)
- static FlyCapture2::PixelFormat [translate](#) ([PixelFormat](#) pixelFormat)
- static [BayerTileFormat](#) [translate](#) (FlyCapture2::BayerTileFormat bayerFormat)
- static FlyCapture2::BayerTileFormat [translate](#) ([BayerTileFormat](#) bayerFormat)
- static [Mode](#) [translate](#) (FlyCapture2::Mode mode)
- static FlyCapture2::Mode [translate](#) ([Mode](#) mode)
- static [BusSpeed](#) [translate](#) (FlyCapture2::BusSpeed busSpeed)
- static FlyCapture2::BusSpeed [translate](#) ([BusSpeed](#) busSpeed)
- static [PCleBusSpeed](#) [translate](#) (FlyCapture2::PCleBusSpeed pcieBusSpeed)
- static FlyCapture2::PCleBusSpeed [translate](#) ([PCleBusSpeed](#) pcieBusSpeed)
- static [ColorProcessingAlgorithm](#) [translate](#) (FlyCapture2::ColorProcessingAlgorithm algorithm)
- static FlyCapture2::ColorProcessingAlgorithm [translate](#) ([ColorProcessingAlgorithm](#) algorithm)
- static [ImageFileFormat](#) [translate](#) (FlyCapture2::ImageFileFormat fileFmt)
- static FlyCapture2::ImageFileFormat [translate](#) ([ImageFileFormat](#) fileFmt)
- static [TiffOption::CompressionMethod](#) [translate](#) (FlyCapture2::TiffOption::CompressionMethod method)
- static FlyCapture2::TiffOption::CompressionMethod [translate](#) ([TiffOption::CompressionMethod](#) method)
- static [StatisticsChannel](#) [translate](#) (FlyCapture2::ImageStatistics::StatisticsChannel channel)
- static FlyCapture2::ImageStatistics::StatisticsChannel [translate](#) ([StatisticsChannel](#) channel)
- static [OSType](#) [translate](#) (FlyCapture2::OSType osType)
- static FlyCapture2::OSType [translate](#) ([OSType](#) osType)
- static [ByteOrder](#) [translate](#) (FlyCapture2::ByteOrder byteOrder)
- static FlyCapture2::ByteOrder [translate](#) ([ByteOrder](#) byteOrder)
- static [GigEPropertyType](#) [translate](#) (FlyCapture2::GigEPropertyType propType)
- static FlyCapture2::GigEPropertyType [translate](#) ([GigEPropertyType](#) propType)
- static void [ToMgd](#) (FlyCapture2::FC2Config *pNative, [FC2Config](#)^ mgd)
- static void [ToNative](#) ([FC2Config](#)^ mgd, FlyCapture2::FC2Config *pNative)
- static void [ToMgd](#) (FlyCapture2::PropertyInfo *pNative, [CameraPropertyInfo](#)^ mgd)
- static void [ToNative](#) ([CameraPropertyInfo](#)^ mgd, FlyCapture2::PropertyInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::Property *pNative, [CameraProperty](#)^ mgd)
- static void [ToNative](#) ([CameraProperty](#)^ mgd, FlyCapture2::Property *pNative)
- static void [ToMgd](#) (FlyCapture2::TriggerModelInfo *pNative, [TriggerModelInfo](#)^ mgd)
- static void [ToNative](#) ([TriggerModelInfo](#)^ mgd, FlyCapture2::TriggerModelInfo *pNative)

- static void [ToMgd](#) (FlyCapture2::TriggerMode *pNative, [TriggerMode](#)^ mgd)
- static void [ToNative](#) ([TriggerMode](#)^ mgd, FlyCapture2::TriggerMode *pNative)
- static void [ToMgd](#) (FlyCapture2::StrobeInfo *pNative, [StrobeInfo](#)^ mgd)
- static void [ToNative](#) ([StrobeInfo](#)^ mgd, FlyCapture2::StrobeInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::StrobeControl *pNative, [StrobeControl](#)^ mgd)
- static void [ToNative](#) ([StrobeControl](#)^ mgd, FlyCapture2::StrobeControl *pNative)
- static void [ToMgd](#) (FlyCapture2::Format7ImageSettings *pNative, [Format7-ImageSettings](#)^ mgd)
- static void [ToNative](#) ([Format7ImageSettings](#)^ mgd, FlyCapture2::Format7ImageSettings *pNative)
- static void [ToMgd](#) (FlyCapture2::Format7Info *pNative, [Format7Info](#)^ mgd)
- static void [ToNative](#) ([Format7Info](#)^ mgd, FlyCapture2::Format7Info *pNative)
- static void [ToMgd](#) (FlyCapture2::Format7PacketInfo *pNative, [Format7Packet-Info](#)^ mgd)
- static void [ToNative](#) ([Format7PacketInfo](#)^ mgd, FlyCapture2::Format7PacketInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::TimeStamp *pNative, [TimeStamp](#)^ mgd)
- static void [ToNative](#) ([TimeStamp](#)^ mgd, FlyCapture2::TimeStamp *pNative)
- static void [ToMgd](#) (FlyCapture2::ConfigROM *pNative, [ConfigROM](#)^ mgd)
- static void [ToNative](#) ([ConfigROM](#)^ mgd, FlyCapture2::ConfigROM *pNative)
- static void [ToMgd](#) (FlyCapture2::CameraInfo *pNative, [CameraInfo](#)^ mgd)
- static void [ToNative](#) ([CameraInfo](#)^ mgd, FlyCapture2::CameraInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::ImageMetadata *pNative, [ImageMetadata](#)^ mgd)
- static void [ToNative](#) ([ImageMetadata](#)^ mgd, FlyCapture2::ImageMetadata *pNative)
- static void [ToMgd](#) (FlyCapture2::LUTData *pNative, [LutData](#)^ mgd)
- static void [ToNative](#) ([LutData](#)^ mgd, FlyCapture2::LUTData *pNative)
- static void [ToMgd](#) (FlyCapture2::EmbeddedImageInfoProperty *pNative, [EmbeddedImageInfoProperty](#)^ mgd)
- static void [ToNative](#) ([EmbeddedImageInfoProperty](#)^ mgd, FlyCapture2::EmbeddedImageInfoProperty *pNative)
- static void [ToMgd](#) (FlyCapture2::EmbeddedImageInfo *pNative, [Embedded-ImageInfo](#)^ mgd)
- static void [ToNative](#) ([EmbeddedImageInfo](#)^ mgd, FlyCapture2::EmbeddedImageInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::PNGOption *pNative, [PngOption](#)^ mgd)
- static void [ToNative](#) ([PngOption](#)^ mgd, FlyCapture2::PNGOption *pNative)
- static void [ToMgd](#) (FlyCapture2::PPMOption *pNative, [PpmOption](#)^ mgd)
- static void [ToNative](#) ([PpmOption](#)^ mgd, FlyCapture2::PPMOption *pNative)
- static void [ToMgd](#) (FlyCapture2::PGMOption *pNative, [PgmOption](#)^ mgd)
- static void [ToNative](#) ([PgmOption](#)^ mgd, FlyCapture2::PGMOption *pNative)
- static void [ToMgd](#) (FlyCapture2::TIFFOption *pNative, [TiffOption](#)^ mgd)
- static void [ToNative](#) ([TiffOption](#)^ mgd, FlyCapture2::TIFFOption *pNative)
- static void [ToMgd](#) (FlyCapture2::JPEGOOption *pNative, [JpegOption](#)^ mgd)
- static void [ToNative](#) ([JpegOption](#)^ mgd, FlyCapture2::JPEGOOption *pNative)
- static void [ToMgd](#) (FlyCapture2::JPG2Option *pNative, [Jpg2Option](#)^ mgd)

- static void [ToNative](#) ([Jpg2Option](#)^ mgd, FlyCapture2::JPG2Option *pNative)
- static void [ToNative](#) ([AviOption](#)^ mgd, FlyCapture2::AVIOption *pNative)
- static void [ToNative](#) ([MJPGOption](#)^ mgd, FlyCapture2::MJPGOption *pNative)
- static void [ToNative](#) ([H264Option](#)^ mgd, FlyCapture2::H264Option *pNative)
- static void [ToMgd](#) (FlyCapture2::SystemInfo *pNative, [SystemInfo](#)^ mgd)
- static void [ToMgd](#) (FlyCapture2::FC2Version *pNative, [FC2Version](#)^ mgd)
- static void [ToMgd](#) (FlyCapture2::IPAddress *pNative, System::Net::IPAddress^ %mgd)
- static void [ToNative](#) (System::Net::IPAddress^ mgd, FlyCapture2::IPAddress *pNative)
- static void [ToMgd](#) (FlyCapture2::MACAddress *pNative, System::Net::NetworkInformation::PhysicalAddress^ %mgd)
- static void [ToNative](#) (System::Net::NetworkInformation::PhysicalAddress^ mgd, - FlyCapture2::MACAddress *pNative)
- static void [ToMgd](#) (FlyCapture2::GigEProperty *pNative, [GigEProperty](#)^ mgd)
- static void [ToNative](#) ([GigEProperty](#)^ mgd, FlyCapture2::GigEProperty *pNative)
- static void [ToMgd](#) (FlyCapture2::GigEImageSettingsInfo *pNative, [GigEImageSettingsInfo](#)^ mgd)
- static void [ToNative](#) ([GigEImageSettingsInfo](#)^ mgd, FlyCapture2::GigEImageSettingsInfo *pNative)
- static void [ToMgd](#) (FlyCapture2::GigEImageSettings *pNative, [GigEImageSettings](#)^ mgd)
- static void [ToNative](#) ([GigEImageSettings](#)^ mgd, FlyCapture2::GigEImageSettings *pNative)
- static void [Translate::ToMgd](#) (FlyCapture2::GigEConfig *pNative, [GigEConfig](#)^ mgd)
- static void [Translate::ToNative](#) ([GigEConfig](#)^ mgd, FlyCapture2::GigEConfig *pNative)
- static void [ToMgd](#) (FlyCapture2::GigEStreamChannel *pNative, [GigEStreamChannel](#)^ mgd)
- static void [ToNative](#) ([GigEStreamChannel](#)^ mgd, FlyCapture2::GigEStreamChannel *pNative)

7.47.1 Member Function Documentation

7.47.1.1 void ToMgd (FlyCapture2::FC2Config * *pNative*, [FC2Config](#)^ *mgd*) [static, package]

7.47.1.2 void ToMgd (FlyCapture2::PropertyInfo * *pNative*, [CameraPropertyInfo](#)^ *mgd*) [static, package]

7.47.1.3 void ToMgd (FlyCapture2::Property * *pNative*, [CameraProperty](#)^ *mgd*) [static, package]

7.47.1.4 void ToMgd (FlyCapture2::TriggerModelInfo * *pNative*, [TriggerModelInfo](#)^ *mgd*) [static, package]

- 7.47.1.5 `void ToMgd (FlyCapture2::TriggerMode * pNative, TriggerMode^ mgd)`
[static, package]
- 7.47.1.6 `void ToMgd (FlyCapture2::StrobeInfo * pNative, StrobeInfo^ mgd)` [static, package]
- 7.47.1.7 `void ToMgd (FlyCapture2::StrobeControl * pNative, StrobeControl^ mgd)`
[static, package]
- 7.47.1.8 `void ToMgd (FlyCapture2::Format7ImageSettings * pNative,
Format7ImageSettings^ mgd)` [static, package]
- 7.47.1.9 `void ToMgd (FlyCapture2::Format7Info * pNative, Format7Info^ mgd)`
[static, package]
- 7.47.1.10 `void ToMgd (FlyCapture2::Format7PacketInfo * pNative, Format7PacketInfo^
mgd)` [static, package]
- 7.47.1.11 `void ToMgd (FlyCapture2::TimeStamp * pNative, TimeStamp^ mgd)`
[static, package]
- 7.47.1.12 `void ToMgd (FlyCapture2::ConfigROM * pNative, ConfigROM^ mgd)`
[static, package]
- 7.47.1.13 `void ToMgd (FlyCapture2::CameraInfo * pNative, CameraInfo^ mgd)`
[static, package]
- 7.47.1.14 `void ToMgd (FlyCapture2::ImageMetadata * pNative, ImageMetadata^ mgd)`
[static, package]
- 7.47.1.15 `void ToMgd (FlyCapture2::LUTData * pNative, LutData^ mgd)` [static, package]
- 7.47.1.16 `void ToMgd (FlyCapture2::EmbeddedImageInfoProperty * pNative,
EmbeddedImageInfoProperty^ mgd)` [static, package]
- 7.47.1.17 `void ToMgd (FlyCapture2::EmbeddedImageInfo * pNative, EmbeddedImageInfo^
mgd)` [static, package]
- 7.47.1.18 `void ToMgd (FlyCapture2::PNGOption * pNative, PngOption^ mgd)`
[static, package]
- 7.47.1.19 `void ToMgd (FlyCapture2::PPMOption * pNative, PpmOption^ mgd)`
[static, package]
- 7.47.1.20 `void ToMgd (FlyCapture2::PGMOption * pNative, PgmOption^ mgd)`
[static, package]

- 7.47.1.21 void ToMgd (FlyCapture2::TIFFOption * *pNative*, TIFFOption^ *mgd*)
[static, package]
- 7.47.1.22 void ToMgd (FlyCapture2::JPEGOption * *pNative*, JpegOption^ *mgd*)
[static, package]
- 7.47.1.23 void ToMgd (FlyCapture2::JPG2Option * *pNative*, Jpg2Option^ *mgd*)
[static, package]
- 7.47.1.24 void ToMgd (FlyCapture2::SystemInfo * *pNative*, SystemInfo^ *mgd*)
[static, package]
- 7.47.1.25 void ToMgd (FlyCapture2::FC2Version * *pNative*, FC2Version^ *mgd*)
[static, package]
- 7.47.1.26 void ToMgd (FlyCapture2::IPAddress * *pNative*, System::Net::IPAddress^ % *mgd*)
[static, package]
- 7.47.1.27 void ToMgd (FlyCapture2::MACAddress * *pNative*, System::Net::-
NetworkInformation::PhysicalAddress^ % *mgd*) [static,
package]
- 7.47.1.28 void ToMgd (FlyCapture2::GigEProperty * *pNative*, GigEProperty^ *mgd*)
[static, package]
- 7.47.1.29 void ToMgd (FlyCapture2::GigEImageSettingsInfo * *pNative*,
GigEImageSettingsInfo^ *mgd*) [static, package]
- 7.47.1.30 void ToMgd (FlyCapture2::GigEImageSettings * *pNative*, GigEImageSettings^
mgd) [static, package]
- 7.47.1.31 void ToMgd (FlyCapture2::GigEStreamChannel * *pNative*, GigEStreamChannel^
mgd) [static, package]
- 7.47.1.32 void ToNative (FC2Config^ *mgd*, FlyCapture2::FC2Config * *pNative*)
[static, package]
- 7.47.1.33 void ToNative (CameraPropertyInfo^ *mgd*, FlyCapture2::PropertyInfo * *pNative*)
[static, package]
- 7.47.1.34 void ToNative (CameraProperty^ *mgd*, FlyCapture2::Property * *pNative*)
[static, package]
- 7.47.1.35 void ToNative (TriggerModelInfo^ *mgd*, FlyCapture2::TriggerModelInfo * *pNative*)
[static, package]
- 7.47.1.36 void ToNative (TriggerMode^ *mgd*, FlyCapture2::TriggerMode * *pNative*)
[static, package]

- 7.47.1.37 `void ToNative (StrobelInfo^ mgd, FlyCapture2::StrobelInfo * pNative)`
[static, package]
- 7.47.1.38 `void ToNative (StrobeControl^ mgd, FlyCapture2::StrobeControl * pNative)`
[static, package]
- 7.47.1.39 `void ToNative (Format7ImageSettings^ mgd,
FlyCapture2::Format7ImageSettings * pNative)` [static, package]
- 7.47.1.40 `void ToNative (Format7Info^ mgd, FlyCapture2::Format7Info * pNative)`
[static, package]
- 7.47.1.41 `void ToNative (Format7PacketInfo^ mgd, FlyCapture2::Format7PacketInfo *
pNative)` [static, package]
- 7.47.1.42 `void ToNative (TimeStamp^ mgd, FlyCapture2::TimeStamp * pNative)`
[static, package]
- 7.47.1.43 `void ToNative (ConfigROM^ mgd, FlyCapture2::ConfigROM * pNative)`
[static, package]
- 7.47.1.44 `void ToNative (CameraInfo^ mgd, FlyCapture2::CameraInfo * pNative)`
[static, package]
- 7.47.1.45 `void ToNative (ImageMetadata^ mgd, FlyCapture2::ImageMetadata * pNative)`
[static, package]
- 7.47.1.46 `void ToNative (LutData^ mgd, FlyCapture2::LUTData * pNative)` [static,
package]
- 7.47.1.47 `void ToNative (EmbeddedImageInfoProperty^ mgd,
FlyCapture2::EmbeddedImageInfoProperty * pNative)` [static, package]
- 7.47.1.48 `void ToNative (EmbeddedImageInfo^ mgd, FlyCapture2::EmbeddedImageInfo *
pNative)` [static, package]
- 7.47.1.49 `void ToNative (PngOption^ mgd, FlyCapture2::PNGOption * pNative)`
[static, package]
- 7.47.1.50 `void ToNative (PpmOption^ mgd, FlyCapture2::PPMOption * pNative)`
[static, package]
- 7.47.1.51 `void ToNative (PgmOption^ mgd, FlyCapture2::PGMOption * pNative)`
[static, package]
- 7.47.1.52 `void ToNative (TiffOption^ mgd, FlyCapture2::TIFFOption * pNative)`
[static, package]

- 7.47.1.53 `void ToNative (JpegOption^ mgd, FlyCapture2::JPEGOption * pNative)`
[static, package]
- 7.47.1.54 `void ToNative (Jpg2Option^ mgd, FlyCapture2::JPG2Option * pNative)`
[static, package]
- 7.47.1.55 `void ToNative (AviOption^ mgd, FlyCapture2::AVIOption * pNative)`
[static, package]
- 7.47.1.56 `void ToNative (MJPGOption^ mgd, FlyCapture2::MJPGOption * pNative)`
[static, package]
- 7.47.1.57 `void ToNative (H264Option^ mgd, FlyCapture2::H264Option * pNative)`
[static, package]
- 7.47.1.58 `void ToNative (System::Net::IPAddress^ mgd, FlyCapture2::IPAddress * pNative)`
[static, package]
- 7.47.1.59 `void ToNative (System::Net::NetworkInformation::PhysicalAddress^ mgd, FlyCapture2::MACAddress * pNative)` [static, package]
- 7.47.1.60 `void ToNative (GigEProperty^ mgd, FlyCapture2::GigEProperty * pNative)`
[static, package]
- 7.47.1.61 `void ToNative (GigEImageSettingsInfo^ mgd, FlyCapture2::GigEImageSettingsInfo * pNative)` [static, package]
- 7.47.1.62 `void ToNative (GigEImageSettings^ mgd, FlyCapture2::GigEImageSettings * pNative)` [static, package]
- 7.47.1.63 `void ToNative (GigEStreamChannel^ mgd, FlyCapture2::GigEStreamChannel * pNative)` [static, package]
- 7.47.1.64 `ErrorType translate (FlyCapture2::ErrorType errorType)` [static, package]
- 7.47.1.65 `FlyCapture2::ErrorType translate (ErrorType errorType)` [static, package]
- 7.47.1.66 `GrabMode translate (FlyCapture2::GrabMode grabMode)` [static, package]
- 7.47.1.67 `FlyCapture2::GrabMode translate (GrabMode grabMode)` [static, package]
- 7.47.1.68 `BandwidthAllocation translate (FlyCapture2::BandwidthAllocation bandwidthAllocation)` [static, package]

- 7.47.1.69 **FlyCapture2::BandwidthAllocation** translate (**BandwidthAllocation** *bandwidthAllocation*) [static, package]
- 7.47.1.70 **InterfaceType** translate (**FlyCapture2::InterfaceType** *interfaceType*) [static, package]
- 7.47.1.71 **FlyCapture2::InterfaceType** translate (**InterfaceType** *interfaceType*) [static, package]
- 7.47.1.72 **DriverType** translate (**FlyCapture2::DriverType** *driverType*) [static, package]
- 7.47.1.73 **FlyCapture2::DriverType** translate (**DriverType** *driverType*) [static, package]
- 7.47.1.74 **PropertyType** translate (**FlyCapture2::PropertyType** *propertyType*) [static, package]
- 7.47.1.75 **FlyCapture2::PropertyType** translate (**PropertyType** *propertyType*) [static, package]
- 7.47.1.76 **FrameRate** translate (**FlyCapture2::FrameRate** *frmRate*) [static, package]
- 7.47.1.77 **FlyCapture2::FrameRate** translate (**FrameRate** *frmRate*) [static, package]
- 7.47.1.78 **VideoMode** translate (**FlyCapture2::VideoMode** *videoMode*) [static, package]
- 7.47.1.79 **FlyCapture2::VideoMode** translate (**VideoMode** *videoMode*) [static, package]
- 7.47.1.80 **PixelFormat** translate (**FlyCapture2::PixelFormat** *pixelFormat*) [static, package]
- 7.47.1.81 **FlyCapture2::PixelFormat** translate (**PixelFormat** *pixelFormat*) [static, package]
- 7.47.1.82 **BayerTileFormat** translate (**FlyCapture2::BayerTileFormat** *bayerFormat*) [static, package]
- 7.47.1.83 **FlyCapture2::BayerTileFormat** translate (**BayerTileFormat** *bayerFormat*) [static, package]
- 7.47.1.84 **Mode** translate (**FlyCapture2::Mode** *mode*) [static, package]
- 7.47.1.85 **FlyCapture2::Mode** translate (**Mode** *mode*) [static, package]

- 7.47.1.86 **BusSpeed** translate (FlyCapture2::BusSpeed *busSpeed*) [static, package]
- 7.47.1.87 FlyCapture2::BusSpeed translate (BusSpeed *busSpeed*) [static, package]
- 7.47.1.88 **PCleBusSpeed** translate (FlyCapture2::PCleBusSpeed *pcieBusSpeed*) [static, package]
- 7.47.1.89 FlyCapture2::PCleBusSpeed translate (PCleBusSpeed *pcieBusSpeed*) [static, package]
- 7.47.1.90 **ColorProcessingAlgorithm** translate (FlyCapture2::ColorProcessingAlgorithm *algorithm*) [static, package]
- 7.47.1.91 FlyCapture2::ColorProcessingAlgorithm translate (ColorProcessingAlgorithm *algorithm*) [static, package]
- 7.47.1.92 **ImageFileFormat** translate (FlyCapture2::ImageFileFormat *fileFmt*) [static, package]
- 7.47.1.93 FlyCapture2::ImageFileFormat translate (ImageFileFormat *fileFmt*) [static, package]
- 7.47.1.94 **TiffOption::CompressionMethod** translate (FlyCapture2::TiffOption::CompressionMethod *method*) [static, package]
- 7.47.1.95 FlyCapture2::TiffOption::CompressionMethod translate (TiffOption::CompressionMethod *method*) [static, package]
- 7.47.1.96 **StatisticsChannel** translate (FlyCapture2::ImageStatistics::StatisticsChannel *channel*) [static, package]
- 7.47.1.97 FlyCapture2::ImageStatistics::StatisticsChannel translate (StatisticsChannel *channel*) [static, package]
- 7.47.1.98 **OSType** translate (FlyCapture2::OSType *osType*) [static, package]
- 7.47.1.99 FlyCapture2::OSType translate (OSType *osType*) [static, package]
- 7.47.1.100 **ByteOrder** translate (FlyCapture2::ByteOrder *byteOrder*) [static, package]
- 7.47.1.101 FlyCapture2::ByteOrder translate (ByteOrder *byteOrder*) [static, package]

- 7.47.1.102 **GigEPropertyType** translate (**FlyCapture2::GigEPropertyType** *propType*)
[static, package]
- 7.47.1.103 **FlyCapture2::GigEPropertyType** translate (**GigEPropertyType** *propType*)
[static, package]
- 7.47.1.104 static void Translate::ToMgd (**FlyCapture2::GigEConfig** * *pNative*, **GigEConfig** ^
mgd) [static, package]
- 7.47.1.105 static void Translate::ToNative (**GigEConfig** ^ *mgd*, **FlyCapture2::GigEConfig** *
pNative) [static, package]

7.48 TriggerMode Struct Reference

A camera trigger.

Properties

- bool **onOff**
Flag controlling on/off.
- unsigned int **polarity**
Polarity value.
- unsigned int **source**
Source value.
- unsigned int **mode**
Mode value.
- unsigned int **parameter**
Parameter value.

7.48.1 Detailed Description

A camera trigger.

7.48.2 Property Documentation

7.48.2.1 unsigned int mode

Mode value.

7.48.2.2 bool onOff

Flag controlling on/off.

7.48.2.3 unsigned int parameter

Parameter value.

7.48.2.4 unsigned int polarity

Polarity value.

7.48.2.5 unsigned int source

Source value.

7.49 TriggerModelInfo Struct Reference

Information about a camera trigger property.

Properties

- bool [present](#)
Presence of trigger mode.
- bool [readOutSupported](#)
Flag indicating if trigger value can be read out.
- bool [onOffSupported](#)
Flag indicating if on/off is supported.
- bool [polaritySupported](#)
Flag indicating if polarity is supported.
- bool [valueReadable](#)
Flag indicating if the value is readable.
- unsigned int [sourceMask](#)
Source mask.
- bool [softwareTriggerSupported](#)
Flag indicating if software trigger is supported.
- unsigned int [modeMask](#)
Mode mask.

7.49.1 Detailed Description

Information about a camera trigger property.

7.49.2 Property Documentation

7.49.2.1 unsigned int modeMask

Mode mask.

7.49.2.2 bool onOffSupported

Flag indicating if on/off is supported.

7.49.2.3 bool polaritySupported

Flag indicating if polarity is supported.

7.49.2.4 bool present

Presence of trigger mode.

7.49.2.5 bool readOutSupported

Flag indicating if trigger value can be read out.

7.49.2.6 bool softwareTriggerSupported

Flag indicating if software trigger is supported.

7.49.2.7 unsigned int sourceMask

Source mask.

7.49.2.8 bool valueReadable

Flag indicating if the value is readable.

Index

- ~CameraControlDialog
 - FlyCapture2Managed::Gui::CameraControlDialog, [34](#)
- ~CameraSelectionDialog
 - FlyCapture2Managed::Gui::CameraSelectionDialog, [44](#)
- ~FC2Exception
 - FlyCapture2Managed::FC2Exception, [52](#)
- ~ManagedAVIRecorder
 - FlyCapture2Managed::ManagedAVIRecorder, [70](#)
- ~ManagedBusManager
 - FlyCapture2Managed::ManagedBusManager, [73](#)
- ~ManagedCamera
 - FlyCapture2Managed::ManagedCamera, [83](#)
- ~ManagedCameraBase
 - FlyCapture2Managed::ManagedCameraBase, [92](#)
- ~ManagedGCCamera
 - FlyCapture2Managed::ManagedGCCamera, [108](#)
- ~ManagedGCPort
 - FlyCapture2Managed::ManagedGCPort, [109](#)
- ~ManagedGigECamera
 - FlyCapture2Managed::ManagedGigECamera, [113](#)
- ~ManagedImage
 - FlyCapture2Managed::ManagedImage, [122](#)
- ~ManagedImageStatistics
 - FlyCapture2Managed::ManagedImageStatistics, [128](#)
- ~ManagedTopologyNode
 - FlyCapture2Managed::ManagedTopologyNode, [132](#)
- AdobeDeflate
 - FlyCapture2Managed::TiffOption, [143](#)
- Any
 - Enumerations, [14](#)
- Arrival
 - Enumerations, [19](#)
- AutoExposure
 - Enumerations, [22](#)
- BGGR
 - Enumerations, [13](#)
- BigEndian
 - Enumerations, [14](#)
- Blue
 - Enumerations, [22](#)
- Bmp
 - Enumerations, [18](#)
- Brightness
 - Enumerations, [22](#)
- BufferFrames
 - Enumerations, [17](#)
- BufferTooSmall
 - Enumerations, [16](#)
- Bus
 - FlyCapture2Managed::ManagedTopologyNode, [132](#)
- BusMasterFailed
 - Enumerations, [16](#)
- BusReset
 - Enumerations, [19](#)
- Camera
 - FlyCapture2Managed::ManagedTopologyNode, [132](#)
- CcittFax3
 - FlyCapture2Managed::TiffOption, [143](#)
- CcittFax4
 - FlyCapture2Managed::TiffOption, [143](#)
- Computer
 - FlyCapture2Managed::ManagedTopologyNode, [132](#)
- ConnectedToChild

- FlyCapture2Managed::Managed-TopologyNode, [132](#)
- ConnectedToParent
 - FlyCapture2Managed::Managed-TopologyNode, [132](#)
- Default
 - Enumerations, [15](#)
- Deflate
 - FlyCapture2Managed::TiffOption, [143](#)
- Directional
 - Enumerations, [15](#)
- DropFrames
 - Enumerations, [17](#)
- EdgeSensing
 - Enumerations, [15](#)
- Enumerations
 - Any, [14](#)
 - Arrival, [19](#)
 - AutoExposure, [22](#)
 - BGGR, [13](#)
 - BigEndian, [14](#)
 - Blue, [22](#)
 - Bmp, [18](#)
 - Brightness, [22](#)
 - BufferFrames, [17](#)
 - BufferTooSmall, [16](#)
 - BusMasterFailed, [16](#)
 - BusReset, [19](#)
 - Default, [15](#)
 - Directional, [15](#)
 - DropFrames, [17](#)
 - EdgeSensing, [15](#)
 - Failed, [15](#)
 - FailedBusMasterConnection, [16](#)
 - FailedGuid, [16](#)
 - Fastest, [14](#)
 - Focus, [22](#)
 - FrameRate, [22](#)
 - FrameRate120, [17](#)
 - FrameRate15, [17](#)
 - FrameRate1_875, [17](#)
 - FrameRate240, [17](#)
 - FrameRate30, [17](#)
 - FrameRate3_75, [17](#)
 - FrameRate60, [17](#)
 - FrameRate7_5, [17](#)
 - FrameRateFormat7, [17](#)
 - FromFileExtension, [18](#)
 - GBRG, [13](#)
 - GRBG, [13](#)
 - Gain, [22](#)
 - Gamma, [22](#)
 - GigE, [19](#)
 - GigE_10000Base_T, [14](#)
 - GigE_1000Base_T, [14](#)
 - GigE_100Base_T, [14](#)
 - GigE_10Base_T, [14](#)
 - GigE_Filter, [15](#)
 - GigE_None, [15](#)
 - GigE_Pro, [15](#)
 - Green, [22](#)
 - Grey, [22](#)
 - HQLinear, [15](#)
 - Heartbeat, [17](#)
 - HeartbeatTimeout, [17](#)
 - Hue, [22](#)
 - IPP, [15](#)
 - ieee1394, [19](#)
 - ieee1394_Cam, [15](#)
 - ieee1394_Juju, [15](#)
 - ieee1394_Pro, [15](#)
 - ieee1394_Raw1394, [15](#)
 - ieee1394_Video1394, [15](#)
 - lidcFailed, [16](#)
 - ImageConsistencyError, [16](#)
 - ImageConversionFailed, [16](#)
 - ImageLibraryFailure, [16](#)
 - Infinite, [18](#)
 - InitFailed, [16](#)
 - InvalidBuManager, [16](#)
 - InvalidGeneration, [16](#)
 - InvalidMode, [16](#)
 - InvalidPacketSize, [16](#)
 - InvalidParameter, [16](#)
 - InvalidSettings, [16](#)
 - Iris, [22](#)
 - IsochAlreadyStarted, [16](#)
 - IsochBandwidthExceeded, [16](#)
 - IsochFailed, [16](#)
 - IsochNotStarted, [16](#)
 - IsochRetrieveBufferFailed, [16](#)
 - IsochStartFailed, [16](#)
 - IsochStopFailed, [16](#)
 - IsochSyncFailed, [16](#)
 - Jpeg, [18](#)
 - Jpeg2000, [18](#)
 - Lightness, [22](#)
 - LinuxX64, [20](#)
 - LinuxX86, [20](#)

LittleEndian, [14](#)
LowLevelFailure, [16](#)
LutFailed, [16](#)
Mac, [20](#)
MemoryAllocationFailed, [16](#)
Mode0, [19](#)
Mode1, [19](#)
Mode10, [19](#)
Mode11, [19](#)
Mode12, [20](#)
Mode13, [20](#)
Mode14, [20](#)
Mode15, [20](#)
Mode16, [20](#)
Mode17, [20](#)
Mode18, [20](#)
Mode19, [20](#)
Mode2, [19](#)
Mode20, [20](#)
Mode21, [20](#)
Mode22, [20](#)
Mode23, [20](#)
Mode24, [20](#)
Mode25, [20](#)
Mode26, [20](#)
Mode27, [20](#)
Mode28, [20](#)
Mode29, [20](#)
Mode3, [19](#)
Mode30, [20](#)
Mode31, [20](#)
Mode4, [19](#)
Mode5, [19](#)
Mode6, [19](#)
Mode7, [19](#)
Mode8, [19](#)
Mode9, [19](#)
NearestNeighbor, [15](#)
NoColorProcessing, [15](#)
None, [13](#), [18](#)
NotConnected, [16](#)
NotFound, [16](#)
NotImplemented, [16](#)
NotInFormat7, [16](#)
NotInitialized, [16](#)
NotSupported, [16](#)
NumberOfFrameRates, [17](#)
NumberOfModes, [20](#)
NumberOfPixelFormats, [21](#)
NumberOfStatisticsChannels, [22](#)
NumberOfVideoModes, [23](#)
Off, [13](#)
Ok, [15](#)
On, [13](#)
PacketDelay, [17](#)
PacketSize, [17](#)
Pan, [22](#)
Pgm, [18](#)
PixelFormat411Yuv8, [21](#)
PixelFormat422Yuv8, [21](#)
PixelFormat422Yuv8Jpeg, [21](#)
PixelFormat444Yuv8, [21](#)
PixelFormatBgr, [21](#)
PixelFormatBgr16, [21](#)
PixelFormatBgru, [21](#)
PixelFormatBgru16, [21](#)
PixelFormatMono12, [21](#)
PixelFormatMono16, [21](#)
PixelFormatMono8, [21](#)
PixelFormatRaw12, [21](#)
PixelFormatRaw16, [21](#)
PixelFormatRaw8, [21](#)
PixelFormatRgb, [21](#)
PixelFormatRgb16, [21](#)
PixelFormatRgb8, [21](#)
PixelFormatRgbu, [21](#)
PixelFormatSignedMono16, [21](#)
PixelFormatSignedRgb16, [21](#)
Png, [18](#)
Ppm, [18](#)
PropertyFailed, [16](#)
PropertyNotPresent, [16](#)
RGGB, [13](#)
Raw, [18](#)
ReadRegisterFailed, [16](#)
Red, [22](#)
RegisterFailed, [16](#)
Removal, [19](#)
Rigorous, [15](#)
S100, [14](#)
S1600, [14](#)
S200, [14](#)
S3200, [14](#)
S400, [14](#)
S480, [14](#)
S5000, [14](#)
S800, [14](#)
Saturation, [22](#)
Sharpness, [22](#)
Shutter, [22](#)

- Speed_2_5, [21](#)
- Speed_5_0, [21](#)
- StrobeFailed, [16](#)
- Temperature, [22](#)
- Tiff, [18](#)
- Tilt, [22](#)
- Timeout, [16](#)
- TriggerDelay, [22](#)
- TriggerFailed, [16](#)
- TriggerMode, [22](#)
- Undefined, [15](#)
- Unknown, [14](#), [15](#), [19](#), [21](#)
- UnknownOS, [20](#)
- Unspecified, [13](#), [18](#), [22](#)
- Unsupported, [13](#)
- Usb2, [19](#)
- Usb3, [19](#)
- Usb3_Pro, [15](#)
- Usb_Cam, [15](#)
- Usb_None, [15](#)
- VideoMode1024x768Rgb, [23](#)
- VideoMode1024x768Y16, [23](#)
- VideoMode1024x768Y8, [23](#)
- VideoMode1024x768Yuv422, [23](#)
- VideoMode1280x960Rgb, [23](#)
- VideoMode1280x960Y16, [23](#)
- VideoMode1280x960Y8, [23](#)
- VideoMode1280x960Yuv422, [23](#)
- VideoMode1600x1200Rgb, [23](#)
- VideoMode1600x1200Y16, [23](#)
- VideoMode1600x1200Y8, [23](#)
- VideoMode1600x1200Yuv422, [23](#)
- VideoMode160x120Yuv444, [23](#)
- VideoMode320x240Yuv422, [23](#)
- VideoMode640x480Rgb, [23](#)
- VideoMode640x480Y16, [23](#)
- VideoMode640x480Y8, [23](#)
- VideoMode640x480Yuv411, [23](#)
- VideoMode640x480Yuv422, [23](#)
- VideoMode800x600Rgb, [23](#)
- VideoMode800x600Y16, [23](#)
- VideoMode800x600Y8, [23](#)
- VideoMode800x600Yuv422, [23](#)
- VideoModeFormat7, [23](#)
- WhiteBalance, [22](#)
- WindowsX64, [20](#)
- WindowsX86, [20](#)
- WriteRegisterFailed, [16](#)
- Zoom, [22](#)
- Failed
 - Enumerations, [15](#)
 - FailedBusMasterConnection
 - Enumerations, [16](#)
 - FailedGuid
 - Enumerations, [16](#)
 - Fastest
 - Enumerations, [14](#)
 - FlyCapture2Managed::ManagedTopology-Node
 - Bus, [132](#)
 - Camera, [132](#)
 - Computer, [132](#)
 - ConnectedToChild, [132](#)
 - ConnectedToParent, [132](#)
 - Node, [132](#)
 - NotConnected, [132](#)
 - FlyCapture2Managed::TiffOption
 - AdobeDeflate, [143](#)
 - CcittFax3, [143](#)
 - CcittFax4, [143](#)
 - Deflate, [143](#)
 - Jpeg, [143](#)
 - Lzw, [143](#)
 - None, [143](#)
 - PackBits, [143](#)
 - Focus
 - Enumerations, [22](#)
 - FrameRate
 - Enumerations, [22](#)
 - FrameRate120
 - Enumerations, [17](#)
 - FrameRate15
 - Enumerations, [17](#)
 - FrameRate1_875
 - Enumerations, [17](#)
 - FrameRate240
 - Enumerations, [17](#)
 - FrameRate30
 - Enumerations, [17](#)
 - FrameRate3_75
 - Enumerations, [17](#)
 - FrameRate60
 - Enumerations, [17](#)
 - FrameRate7_5
 - Enumerations, [17](#)
 - FrameRateFormat7
 - Enumerations, [17](#)
 - FromFileExtension
 - Enumerations, [18](#)
 - GBRG

- Enumerations, [13](#)
- GRBG
 - Enumerations, [13](#)
- Gain
 - Enumerations, [22](#)
- Gamma
 - Enumerations, [22](#)
- GigE
 - Enumerations, [19](#)
- GigE_10000Base_T
 - Enumerations, [14](#)
- GigE_1000Base_T
 - Enumerations, [14](#)
- GigE_100Base_T
 - Enumerations, [14](#)
- GigE_10Base_T
 - Enumerations, [14](#)
- GigE_Filter
 - Enumerations, [15](#)
- GigE_None
 - Enumerations, [15](#)
- GigE_Pro
 - Enumerations, [15](#)
- Green
 - Enumerations, [22](#)
- Grey
 - Enumerations, [22](#)
- HQLinear
 - Enumerations, [15](#)
- Heartbeat
 - Enumerations, [17](#)
- HeartbeatTimeout
 - Enumerations, [17](#)
- Hue
 - Enumerations, [22](#)
- IPP
 - Enumerations, [15](#)
- leee1394
 - Enumerations, [19](#)
- leee1394_Cam
 - Enumerations, [15](#)
- leee1394_Juju
 - Enumerations, [15](#)
- leee1394_Pro
 - Enumerations, [15](#)
- leee1394_Raw1394
 - Enumerations, [15](#)
- leee1394_Video1394
 - Enumerations, [15](#)
- lidcFailed
 - Enumerations, [16](#)
- ImageConsistencyError
 - Enumerations, [16](#)
- ImageConversionFailed
 - Enumerations, [16](#)
- ImageLibraryFailure
 - Enumerations, [16](#)
- Infinite
 - Enumerations, [18](#)
- InitFailed
 - Enumerations, [16](#)
- InvalidBuManager
 - Enumerations, [16](#)
- InvalidGeneration
 - Enumerations, [16](#)
- InvalidMode
 - Enumerations, [16](#)
- InvalidPacketSize
 - Enumerations, [16](#)
- InvalidParameter
 - Enumerations, [16](#)
- InvalidSettings
 - Enumerations, [16](#)
- Iris
 - Enumerations, [22](#)
- IsochAlreadyStarted
 - Enumerations, [16](#)
- IsochBandwidthExceeded
 - Enumerations, [16](#)
- IsochFailed
 - Enumerations, [16](#)
- IsochNotStarted
 - Enumerations, [16](#)
- IsochRetrieveBufferFailed
 - Enumerations, [16](#)
- IsochStartFailed
 - Enumerations, [16](#)
- IsochStopFailed
 - Enumerations, [16](#)
- IsochSyncFailed
 - Enumerations, [16](#)
- Jpeg
 - Enumerations, [18](#)
 - FlyCapture2Managed::TiffOption, [143](#)
- Jpeg2000
 - Enumerations, [18](#)
- Lightness
 - Enumerations, [22](#)
- LinuxX64

- Enumerations, [20](#)
- LinuxX86
 - Enumerations, [20](#)
- LittleEndian
 - Enumerations, [14](#)
- LowLevelFailure
 - Enumerations, [16](#)
- LutFailed
 - Enumerations, [16](#)
- Lzw
 - FlyCapture2Managed::TiffOption, [143](#)
- Mac
 - Enumerations, [20](#)
- MemoryAllocationFailed
 - Enumerations, [16](#)
- Mode0
 - Enumerations, [19](#)
- Mode1
 - Enumerations, [19](#)
- Mode10
 - Enumerations, [19](#)
- Mode11
 - Enumerations, [19](#)
- Mode12
 - Enumerations, [20](#)
- Mode13
 - Enumerations, [20](#)
- Mode14
 - Enumerations, [20](#)
- Mode15
 - Enumerations, [20](#)
- Mode16
 - Enumerations, [20](#)
- Mode17
 - Enumerations, [20](#)
- Mode18
 - Enumerations, [20](#)
- Mode19
 - Enumerations, [20](#)
- Mode2
 - Enumerations, [19](#)
- Mode20
 - Enumerations, [20](#)
- Mode21
 - Enumerations, [20](#)
- Mode22
 - Enumerations, [20](#)
- Mode23
 - Enumerations, [20](#)
- Mode24
 - Enumerations, [20](#)
- Mode25
 - Enumerations, [20](#)
- Mode26
 - Enumerations, [20](#)
- Mode27
 - Enumerations, [20](#)
- Mode28
 - Enumerations, [20](#)
- Mode29
 - Enumerations, [20](#)
- Mode3
 - Enumerations, [19](#)
- Mode30
 - Enumerations, [20](#)
- Mode31
 - Enumerations, [20](#)
- Mode4
 - Enumerations, [19](#)
- Mode5
 - Enumerations, [19](#)
- Mode6
 - Enumerations, [19](#)
- Mode7
 - Enumerations, [19](#)
- Mode8
 - Enumerations, [19](#)
- Mode9
 - Enumerations, [19](#)
- NearestNeighbor
 - Enumerations, [15](#)
- NoColorProcessing
 - Enumerations, [15](#)
- Node
 - FlyCapture2Managed::Managed-TopologyNode, [132](#)
- None
 - Enumerations, [13](#), [18](#)
 - FlyCapture2Managed::TiffOption, [143](#)
- NotConnected
 - Enumerations, [16](#)
 - FlyCapture2Managed::Managed-TopologyNode, [132](#)
- NotFound
 - Enumerations, [16](#)
- NotImplemented
 - Enumerations, [16](#)
- NotInFormat7

- Enumerations, [16](#)
- NotInitialized
 - Enumerations, [16](#)
- NotSupported
 - Enumerations, [16](#)
- NumberOfFrameRates
 - Enumerations, [17](#)
- NumberOfModes
 - Enumerations, [20](#)
- NumberOfPixelFormats
 - Enumerations, [21](#)
- NumberOfStatisticsChannels
 - Enumerations, [22](#)
- NumberOfVideoModes
 - Enumerations, [23](#)
- Off
 - Enumerations, [13](#)
- Ok
 - Enumerations, [15](#)
- On
 - Enumerations, [13](#)
- PackBits
 - FlyCapture2Managed::TiffOption, [143](#)
- PacketDelay
 - Enumerations, [17](#)
- PacketSize
 - Enumerations, [17](#)
- Pan
 - Enumerations, [22](#)
- Pgm
 - Enumerations, [18](#)
- PixelFormat411Yuv8
 - Enumerations, [21](#)
- PixelFormat422Yuv8
 - Enumerations, [21](#)
- PixelFormat422Yuv8Jpeg
 - Enumerations, [21](#)
- PixelFormat444Yuv8
 - Enumerations, [21](#)
- PixelFormatBgr
 - Enumerations, [21](#)
- PixelFormatBgr16
 - Enumerations, [21](#)
- PixelFormatBgru
 - Enumerations, [21](#)
- PixelFormatBgru16
 - Enumerations, [21](#)
- PixelFormatMono12
 - Enumerations, [21](#)
- PixelFormatMono16
 - Enumerations, [21](#)
- PixelFormatMono8
 - Enumerations, [21](#)
- PixelFormatRaw12
 - Enumerations, [21](#)
- PixelFormatRaw16
 - Enumerations, [21](#)
- PixelFormatRaw8
 - Enumerations, [21](#)
- PixelFormatRgb
 - Enumerations, [21](#)
- PixelFormatRgb16
 - Enumerations, [21](#)
- PixelFormatRgb8
 - Enumerations, [21](#)
- PixelFormatRgba
 - Enumerations, [21](#)
- PixelFormatSignedMono16
 - Enumerations, [21](#)
- PixelFormatSignedRgb16
 - Enumerations, [21](#)
- Png
 - Enumerations, [18](#)
- Ppm
 - Enumerations, [18](#)
- PropertyFailed
 - Enumerations, [16](#)
- PropertyNotPresent
 - Enumerations, [16](#)
- RGGB
 - Enumerations, [13](#)
- Raw
 - Enumerations, [18](#)
- ReadRegisterFailed
 - Enumerations, [16](#)
- Red
 - Enumerations, [22](#)
- RegisterFailed
 - Enumerations, [16](#)
- Removal
 - Enumerations, [19](#)
- Rigorous
 - Enumerations, [15](#)
- S100
 - Enumerations, [14](#)
- S1600
 - Enumerations, [14](#)
- S200
 - Enumerations, [14](#)

- S3200
 - Enumerations, [14](#)
- S400
 - Enumerations, [14](#)
- S480
 - Enumerations, [14](#)
- S5000
 - Enumerations, [14](#)
- S800
 - Enumerations, [14](#)
- Saturation
 - Enumerations, [22](#)
- Sharpness
 - Enumerations, [22](#)
- Shutter
 - Enumerations, [22](#)
- Speed_2_5
 - Enumerations, [21](#)
- Speed_5_0
 - Enumerations, [21](#)
- StrobeFailed
 - Enumerations, [16](#)
- Temperature
 - Enumerations, [22](#)
- Tiff
 - Enumerations, [18](#)
- Tilt
 - Enumerations, [22](#)
- Timeout
 - Enumerations, [16](#)
- TriggerDelay
 - Enumerations, [22](#)
- TriggerFailed
 - Enumerations, [16](#)
- TriggerMode
 - Enumerations, [22](#)
- Undefined
 - Enumerations, [15](#)
- Unknown
 - Enumerations, [14](#), [15](#), [19](#), [21](#)
- UnknownOS
 - Enumerations, [20](#)
- Unspecified
 - Enumerations, [13](#), [18](#), [22](#)
- Unsupported
 - Enumerations, [13](#)
- Usb2
 - Enumerations, [19](#)
- Usb3
 - Enumerations, [19](#)
- Usb3_Pro
 - Enumerations, [15](#)
- Usb_Cam
 - Enumerations, [15](#)
- Usb_None
 - Enumerations, [15](#)
- VideoMode1024x768Rgb
 - Enumerations, [23](#)
- VideoMode1024x768Y16
 - Enumerations, [23](#)
- VideoMode1024x768Y8
 - Enumerations, [23](#)
- VideoMode1024x768Yuv422
 - Enumerations, [23](#)
- VideoMode1280x960Rgb
 - Enumerations, [23](#)
- VideoMode1280x960Y16
 - Enumerations, [23](#)
- VideoMode1280x960Y8
 - Enumerations, [23](#)
- VideoMode1280x960Yuv422
 - Enumerations, [23](#)
- VideoMode1600x1200Rgb
 - Enumerations, [23](#)
- VideoMode1600x1200Y16
 - Enumerations, [23](#)
- VideoMode1600x1200Y8
 - Enumerations, [23](#)
- VideoMode1600x1200Yuv422
 - Enumerations, [23](#)
- VideoMode1600x1200Yuv444
 - Enumerations, [23](#)
- VideoMode320x240Yuv422
 - Enumerations, [23](#)
- VideoMode640x480Rgb
 - Enumerations, [23](#)
- VideoMode640x480Y16
 - Enumerations, [23](#)
- VideoMode640x480Y8
 - Enumerations, [23](#)
- VideoMode640x480Yuv411
 - Enumerations, [23](#)
- VideoMode640x480Yuv422
 - Enumerations, [23](#)
- VideoMode800x600Rgb
 - Enumerations, [23](#)
- VideoMode800x600Y16
 - Enumerations, [23](#)
- VideoMode800x600Y8
 - Enumerations, [23](#)

- VideoMode800x600Yuv422
 - Enumerations, [23](#)
- VideoModeFormat7
 - Enumerations, [23](#)
- WhiteBalance
 - Enumerations, [22](#)
- WindowsX64
 - Enumerations, [20](#)
- WindowsX86
 - Enumerations, [20](#)
- WriteRegisterFailed
 - Enumerations, [16](#)
- Zoom
 - Enumerations, [22](#)
- AVIAppend
 - FlyCapture2Managed::ManagedAVI-Recorder, [70](#)
- AVIClose
 - FlyCapture2Managed::ManagedAVI-Recorder, [70](#)
- AVIOpen
 - FlyCapture2Managed::ManagedAVI-Recorder, [70](#), [71](#)
- AviOption, [33](#)
 - FlyCapture2Managed::AviOption, [33](#)
- BandwidthAllocation
 - Enumerations, [13](#)
- BayerTileFormat
 - Enumerations, [13](#)
- BusSpeed
 - Enumerations, [13](#)
- ByteOrder
 - Enumerations, [14](#)
- CalculateStatistics
 - FlyCapture2Managed::Managed-Image, [122](#)
- CameraControlDialog, [34](#)
 - FlyCapture2Managed::Gui::Camera-ControlDialog, [34](#)
- CameraInfo, [34](#)
- CameraProperty, [39](#)
 - FlyCapture2Managed::Camera-Property, [40](#)
- CameraPropertyInfo, [41](#)
 - FlyCapture2Managed::Camera-PropertyInfo, [42](#)
- CameraSelectionDialog, [44](#)
 - FlyCapture2Managed::Gui::Camera-SelectionDialog, [44](#)
- CauseType
 - FlyCapture2Managed::FC2Exception, [52](#)
- ColorProcessingAlgorithm
 - Enumerations, [14](#)
- CompressionMethod
 - FlyCapture2Managed::TiffOption, [143](#)
- ConfigROM, [45](#)
- Connect
 - FlyCapture2Managed::Gui::Camera-ControlDialog, [34](#)
 - FlyCapture2Managed::Managed-Camera, [83](#)
 - FlyCapture2Managed::Managed-CameraBase, [92](#)
 - FlyCapture2Managed::ManagedGC-Camera, [108](#)
 - FlyCapture2Managed::ManagedGig-ECamera, [113](#)
- Convert
 - FlyCapture2Managed::Managed-Image, [122](#)
- ConvertToManagedGuid
 - FlyCapture2Managed::Managed-BusManager, [74](#)
- ConvertToNativeGuid
 - FlyCapture2Managed::Managed-BusManager, [74](#)
- DetermineBitsPerPixel
 - FlyCapture2Managed::Managed-Image, [122](#)
- DisableAll
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- Disconnect
 - FlyCapture2Managed::Gui::Camera-ControlDialog, [34](#)
 - FlyCapture2Managed::Managed-CameraBase, [92](#)
 - FlyCapture2Managed::ManagedGC-Camera, [108](#)
- DiscoverGigECameras
 - FlyCapture2Managed::Managed-BusManager, [74](#)
- DiscoverGigEPacketSize
 - FlyCapture2Managed::ManagedGig-ECamera, [113](#)
- DriverType
 - Enumerations, [15](#)
- EmbeddedImageInfo, [47](#)

- FlyCapture2Managed::Embedded-ImageInfo, [47](#)
- EmbeddedImageInfoProperty, [48](#)
- EnableAll
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- EnableGreyOnly
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- EnableHSLOnly
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- EnableLUT
 - FlyCapture2Managed::Managed-CameraBase, [92](#)
- EnableRGBOnly
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- EnumCallback
 - FlyCapture2Managed, [32](#)
- Enumerations, [11](#)
 - BandwidthAllocation, [13](#)
 - BayerTileFormat, [13](#)
 - BusSpeed, [13](#)
 - ByteOrder, [14](#)
 - ColorProcessingAlgorithm, [14](#)
 - DriverType, [15](#)
 - ErrorType, [15](#)
 - FrameRate, [16](#)
 - GigEPropertyType, [17](#)
 - GrabMode, [17](#)
 - GrabTimeout, [18](#)
 - ImageFileFormat, [18](#)
 - InterfaceType, [18](#)
 - ManagedCallbackType, [19](#)
 - Mode, [19](#)
 - OSType, [20](#)
 - PCleBusSpeed, [20](#)
 - PixelFormat, [21](#)
 - PropertyType, [21](#)
 - StatisticsChannel, [22](#)
 - VideoMode, [22](#)
- Equals
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- ErrorType
 - Enumerations, [15](#)
- FC2Config, [49](#)
 - FlyCapture2Managed::FC2Config, [50](#)
- FC2Exception, [51](#)
 - FlyCapture2Managed::FC2Exception, [52](#)
- FC2Version, [52](#)
- FireBusReset
 - FlyCapture2Managed::Managed-BusManager, [74](#)
- FireSoftwareTrigger
 - FlyCapture2Managed::Managed-CameraBase, [93](#)
- FlyCapture2, [27](#)
- FlyCapture2Managed, [27](#)
 - EnumCallback, [32](#)
 - ImageCallbackDelegate, [32](#)
 - ImageEventCallback, [32](#)
- FlyCapture2Managed::AviOption
 - AviOption, [33](#)
 - frameRate, [33](#)
- FlyCapture2Managed::CameraInfo
 - applicationIPAddress, [36](#)
 - applicationPort, [36](#)
 - bayerTileFormat, [36](#)
 - busNumber, [36](#)
 - ccpStatus, [37](#)
 - configROM, [37](#)
 - defaultGateway, [37](#)
 - driverName, [37](#)
 - driverType, [37](#)
 - firmwareBuildTime, [37](#)
 - firmwareVersion, [37](#)
 - gigEMajorVersion, [37](#)
 - gigEMinorVersion, [37](#)
 - iidcVersion, [37](#)
 - interfaceType, [38](#)
 - ipAddress, [38](#)
 - isColorCamera, [38](#)
 - macAddress, [38](#)
 - maximumBusSpeed, [38](#)
 - modelName, [38](#)
 - nodeNumber, [38](#)
 - pcieBusSpeed, [38](#)
 - sensorInfo, [38](#)
 - sensorResolution, [38](#)
 - serialNumber, [39](#)
 - subnetMask, [39](#)
 - userDefinedName, [39](#)
 - vendorName, [39](#)
 - xmlURL1, [39](#)
 - xmlURL2, [39](#)
- FlyCapture2Managed::CameraProperty

- CameraProperty, 40
 - absControl, 40
 - absValue, 40
 - autoManualMode, 40
 - onOff, 41
 - onePush, 40
 - present, 41
 - type, 41
 - valueA, 41
 - valueB, 41
- FlyCapture2Managed::CameraProperty-Info
 - CameraPropertyInfo, 42
 - absMax, 42
 - absMin, 42
 - absValSupported, 42
 - autoSupported, 43
 - manualSupported, 43
 - max, 43
 - min, 43
 - onOffSupported, 43
 - onePushSupported, 43
 - present, 43
 - readOutSupported, 43
 - type, 43
 - unitAbbr, 43
 - units, 44
- FlyCapture2Managed::ConfigROM
 - chipIdHi, 46
 - chipIdLo, 46
 - keyword, 46
 - nodeVendorId, 46
 - unitSWVer, 46
 - unitSpecId, 46
 - unitSubSWVer, 46
 - vendorUniqueInfo0, 46
 - vendorUniqueInfo1, 46
 - vendorUniqueInfo2, 46
 - vendorUniqueInfo3, 47
- FlyCapture2Managed::EmbeddedImage-Info
 - EmbeddedImageInfo, 47
 - GPIOPinState, 48
 - ROIPosition, 48
 - brightness, 47
 - exposure, 47
 - frameCounter, 47
 - gain, 48
 - shutter, 48
 - strobePattern, 48
 - timestamp, 48
 - whiteBalance, 48
- FlyCapture2Managed::EmbeddedImage-InfoProperty
 - available, 48
 - onOff, 48
- FlyCapture2Managed::FC2Config
 - FC2Config, 50
 - asyncBusSpeed, 50
 - bandwidthAllocation, 50
 - grabMode, 50
 - grabTimeout, 50
 - isochBusSpeed, 50
 - minNumImageNotifications, 50
 - numBuffers, 50
 - numImageNotifications, 50
 - registerTimeout, 51
 - registerTimeoutRetries, 51
- FlyCapture2Managed::FC2Exception
 - ~FC2Exception, 52
 - CauseType, 52
 - FC2Exception, 52
 - NativeErrorTrace, 52
 - Type, 52
- FlyCapture2Managed::FC2Version
 - build, 53
 - major, 53
 - minor, 53
 - type, 53
- FlyCapture2Managed::Format7Image-Settings
 - height, 54
 - mode, 54
 - offsetX, 54
 - offsetY, 54
 - pixelFormat, 54
 - width, 54
- FlyCapture2Managed::Format7Info
 - imageHStepSize, 55
 - imageVStepSize, 55
 - maxHeight, 55
 - maxPacketSize, 56
 - maxWidth, 56
 - minPacketSize, 56
 - mode, 56
 - offsetHStepSize, 56
 - offsetVStepSize, 56
 - packetSize, 56
 - percentage, 56
 - pixelFormatBitField, 56

- vendorPixelFormatBitField, 56
- FlyCapture2Managed::Format7PacketInfo
 - maxBytesPerPacket, 57
 - recommendedBytesPerPacket, 57
 - unitBytesPerPacket, 57
- FlyCapture2Managed::GigEConfig
 - enablePacketResend, 58
- FlyCapture2Managed::GigEImage-Settings
 - height, 59
 - offsetX, 59
 - offsetY, 59
 - pixelFormat, 59
 - width, 59
- FlyCapture2Managed::GigEImage-SettingsInfo
 - imageHStepSize, 60
 - imageVStepSize, 60
 - maxHeight, 60
 - maxWidth, 60
 - offsetHStepSize, 60
 - offsetVStepSize, 60
 - pixelFormatBitField, 60
 - vendorPixelFormatBitField, 60
- FlyCapture2Managed::GigEProperty
 - isReadable, 61
 - isWritable, 61
 - max, 61
 - min, 61
 - propType, 61
 - value, 62
- FlyCapture2Managed::GigEStream-Channel
 - destinationIpAddress, 62
 - doNotFragment, 62
 - hostPort, 63
 - interPacketDelay, 63
 - networkInterfaceIndex, 63
 - packetSize, 63
 - sourcePort, 63
- FlyCapture2Managed::Gui, 32
- FlyCapture2Managed::Gui::Camera-ControlDialog
 - ~CameraControlDialog, 34
 - CameraControlDialog, 34
 - Connect, 34
 - Disconnect, 34
 - Hide, 34
 - IsVisible, 34
 - SetTitle, 34
 - Show, 34
- FlyCapture2Managed::Gui::Camera-SelectionDialog
 - ~CameraSelectionDialog, 44
 - CameraSelectionDialog, 44
 - GetSelectedCameraGuids, 44
 - SetTitle, 44
 - ShowModal, 45
- FlyCapture2Managed::H264Option
 - H264Option, 64
 - bitrate, 64
 - frameRate, 64
 - height, 64
 - width, 64
- FlyCapture2Managed::ImageMetadata
 - embeddedBrightness, 65
 - embeddedExposure, 65
 - embeddedFrameCounter, 65
 - embeddedGPIOPinState, 65
 - embeddedGain, 65
 - embeddedROIPosition, 65
 - embeddedShutter, 66
 - embeddedStrobePattern, 66
 - embeddedTimeStamp, 66
 - embeddedWhiteBalance, 66
- FlyCapture2Managed::JpegOption
 - JpegOption, 67
 - progressive, 67
 - quality, 67
- FlyCapture2Managed::Jpg2Option
 - Jpg2Option, 67
 - quality, 68
- FlyCapture2Managed::LutData
 - enabled, 68
 - inputBitDepth, 68
 - numBanks, 68
 - numChannels, 69
 - numEntries, 69
 - outputBitDepth, 69
 - supported, 69
- FlyCapture2Managed::MJPGOption
 - MJPGOption, 135
 - frameRate, 135
 - quality, 135
- FlyCapture2Managed::ManagedAVI-Recorder
 - ~ManagedAVIRecorder, 70
 - AVIAppend, 70
 - AVIClose, 70
 - AVIOpen, 70, 71

- ManagedAVIRecorder, 70
- FlyCapture2Managed::ManagedBus-Manager
 - ~ManagedBusManager, 73
 - ConvertToManagedGuid, 74
 - ConvertToNativeGuid, 74
 - DiscoverGigECameras, 74
 - FireBusReset, 74
 - ForceAllIPAddressesAutomatically, 75
 - ForceIPAddressToCamera, 75
 - GetCameraFromIPAddress, 76
 - GetCameraFromIndex, 75
 - GetCameraFromSerialNumber, 76
 - GetCameraSerialNumberFromIndex, 76
 - GetDeviceFromIndex, 77
 - GetInterfaceTypeFromGuid, 77
 - GetNumOfCameras, 77
 - GetNumOfDevices, 77
 - GetTopology, 78
 - GetUsbLinkInfo, 78
 - GetUsbPortStatus, 78
 - IsCameraControlable, 78
 - ManagedBusManager, 73
 - ReadPhyRegister, 79
 - RegisterCallback, 79
 - RescanBus, 79
 - UnregisterCallback, 80
 - WritePhyRegister, 80
- FlyCapture2Managed::ManagedCamera
 - ~ManagedCamera, 83
 - Connect, 83
 - GetFormat7Configuration, 83
 - GetFormat7Info, 84
 - GetVideoModeAndFrameRate, 84
 - GetVideoModeAndFrameRateInfo, 84
 - ManagedCamera, 83
 - SetFormat7Configuration, 85
 - SetVideoModeAndFrameRate, 86
 - StartSyncCapture, 86
 - ValidateFormat7Settings, 87
- FlyCapture2Managed::ManagedCamera-Base
 - ~ManagedCameraBase, 92
 - Connect, 92
 - Disconnect, 92
 - EnableLUT, 92
 - FireSoftwareTrigger, 93
 - GetActiveLUTBank, 93
 - GetCameraInfo, 93
 - GetConfiguration, 93
 - GetCycleTime, 93
 - GetEmbeddedImageInfo, 94
 - GetGPIOPinDirection, 94
 - GetLUTBankInfo, 94
 - GetLUTChannel, 95
 - GetLUTInfo, 95
 - GetMemoryChannel, 95
 - GetMemoryChannelInfo, 96
 - GetNativeCamera, 96
 - GetProperty, 96
 - GetPropertyInfo, 97
 - GetRegisterString, 97
 - GetStrobe, 97
 - GetStrobeInfo, 98
 - GetTriggerMode, 98
 - GetTriggerModelInfo, 98
 - IsConnected, 98
 - ManagedCameraBase, 92
 - OnNativeCallback, 99
 - ReadRegister, 99
 - ReadRegisterBlock, 99
 - RestoreFromMemoryChannel, 99
 - RetrieveBuffer, 100
 - SaveToMemoryChannel, 100
 - SetActiveLUTBank, 101
 - SetCallback, 101
 - SetCamera, 101
 - SetConfiguration, 101
 - SetEmbeddedImageInfo, 101
 - SetGPIOPinDirection, 102
 - SetLUTChannel, 103
 - SetProperty, 103
 - SetStrobe, 104
 - SetTriggerMode, 104
 - StartCapture, 104, 105
 - StopCapture, 105
 - WaitForBufferEvent, 105
 - WriteRegister, 105, 106
 - WriteRegisterBlock, 106
 - m_externalDelegate, 106
 - m_internalDelegate, 106
 - m_isLocal, 107
 - m_pNativeCamBase, 107
- FlyCapture2Managed::ManagedGCC-
Camera
 - ~ManagedGCCamera, 108
 - Connect, 108

- Disconnect, 108
- GetNodeMap, 109
- ManagedGCCamera, 108
- SetCamera, 109
- FlyCapture2Managed::ManagedGCPort
 - ~ManagedGCPort, 109
 - ManagedGCPort, 109
 - Read, 109
 - Write, 109
- FlyCapture2Managed::ManagedGigE-Camera
 - ~ManagedGigECamera, 113
 - Connect, 113
 - DiscoverGigEPacketSize, 113
 - GetGigEConfig, 113
 - GetGigEImageBinningSettings, 114
 - GetGigEImageSettings, 114
 - GetGigEImageSettingsInfo, 114
 - GetGigEImagingMode, 114
 - GetGigEProperty, 114
 - GetGigEStreamChannelInfo, 115
 - GetNumStreamChannels, 115
 - ManagedGigECamera, 113
 - QueryGigEImagingMode, 115
 - ReadGVCPMemory, 115
 - ReadGVCPRegister, 116
 - ReadGVCPRegisterBlock, 116
 - SetGigEConfig, 116
 - SetGigEImageBinningSettings, 116
 - SetGigEImageSettings, 117
 - SetGigEImagingMode, 117
 - SetGigEProperty, 117
 - SetGigEStreamChannelInfo, 117
 - WriteGVCPMemory, 117
 - WriteGVCPRegister, 118
 - WriteGVCPRegisterBlock, 118
- FlyCapture2Managed::ManagedImage
 - ~ManagedImage, 122
 - CalculateStatistics, 122
 - Convert, 122
 - DetermineBitsPerPixel, 122
 - GetNativeImage, 123
 - GetRawNativeImagePointer, 123
 - IsNativeImageValid, 123
 - ManagedImage, 121, 122
 - ReleaseBuffer, 123
 - Save, 123, 124
 - SetData, 125
 - SetDimensions, 125
 - bayerTileFormat, 125
 - bitmap, 125
 - bitsPerPixel, 125
 - colorProcessingAlgorithm, 126
 - cols, 126
 - data, 126
 - defaultColorProcessingAlgorithm, 126
 - defaultOutputPixelFormat, 126
 - imageMetadata, 126
 - pixelFormat, 126
 - receivedDataSize, 126
 - rows, 126
 - stride, 127
 - timeStamp, 127
- FlyCapture2Managed::ManagedImage-Statistics
 - ~ManagedImageStatistics, 128
 - DisableAll, 128
 - EnableAll, 128
 - EnableGreyOnly, 128
 - EnableHSLOnly, 128
 - EnableRGBOnly, 128
 - GetChannelStatus, 128
 - GetHistogram, 128
 - GetMean, 128
 - GetNativeImageStatistics, 128
 - GetNumPixelValues, 128
 - GetPixelValueRange, 128
 - GetRange, 128
 - GetStatistics, 128
 - ManagedImageStatistics, 128
 - SetChannelStatus, 128
- FlyCapture2Managed::ManagedPGR-Guid
 - Equals, 130
 - GetHashCode, 130
 - ManagedPGRGuid, 129
 - operator=, 130
 - operator==, 130
 - value0, 130
 - value1, 130
 - value2, 130
 - value3, 130
- FlyCapture2Managed::ManagedTopology-Node
 - ~ManagedTopologyNode, 132
 - GetChild, 132
 - GetDeviceId, 133
 - GetGuid, 133
 - GetInterfaceType, 133

- GetNodeType, 133
- GetNumChildren, 133
- GetNumPorts, 133
- GetPortType, 134
- ManagedTopologyNode, 132
- NodeType, 132
- PortType, 132
- TranslateNodeType, 134
- TranslatePortType, 134
- FlyCapture2Managed::ManagedUtilities
 - LaunchBrowser, 135
 - LaunchCommand, 135
 - LaunchHelp, 135
 - libraryVersion, 135
 - systemInfo, 135
- FlyCapture2Managed::PgmOption
 - PgmOption, 136
 - binaryFile, 136
- FlyCapture2Managed::PngOption
 - PngOption, 137
 - compressionLevel, 137
 - interlaced, 137
- FlyCapture2Managed::PpmOption
 - PpmOption, 138
 - binaryFile, 138
- FlyCapture2Managed::StrobeControl
 - delay, 138
 - duration, 138
 - onOff, 138
 - polarity, 139
 - source, 139
- FlyCapture2Managed::StrobeInfo
 - maxValue, 139
 - minValue, 139
 - onOffSupported, 140
 - polaritySupported, 140
 - present, 140
 - readOutSupported, 140
 - source, 140
- FlyCapture2Managed::SystemInfo
 - byteOrder, 141
 - cpuDescription, 141
 - driverList, 141
 - gpuDescription, 141
 - libraryList, 141
 - numCpuCores, 141
 - osDescription, 141
 - osType, 142
 - screenHeight, 142
 - screenWidth, 142
 - systemMemorySize, 142
- FlyCapture2Managed::TiffOption
 - CompressionMethod, 143
 - TiffOption, 143
 - compression, 143
- FlyCapture2Managed::TimeStamp
 - cycleCount, 144
 - cycleOffset, 144
 - cycleSeconds, 144
 - microSeconds, 144
 - seconds, 144
- FlyCapture2Managed::Translate
 - ToMgd, 147–149
 - ToNative, 149–151
 - Translate::ToMgd, 154
 - Translate::ToNative, 154
 - translate, 151–154
- FlyCapture2Managed::TriggerMode
 - mode, 154
 - onOff, 154
 - parameter, 154
 - polarity, 155
 - source, 155
- FlyCapture2Managed::TriggerModeInfo
 - modeMask, 156
 - onOffSupported, 156
 - polaritySupported, 156
 - present, 156
 - readOutSupported, 156
 - softwareTriggerSupported, 156
 - sourceMask, 156
 - valueReadable, 156
- ForceAllIPAddressesAutomatically
 - FlyCapture2Managed::Managed-
BusManager, 75
- ForceIPAddressToCamera
 - FlyCapture2Managed::Managed-
BusManager, 75
- Format7ImageSettings, 53
- Format7Info, 54
- Format7PacketInfo, 57
- FrameRate
 - Enumerations, 16
- GPIOPinState
 - FlyCapture2Managed::Embedded-
ImageInfo, 48
- GetActiveLUTBank
 - FlyCapture2Managed::Managed-
CameraBase, 93
- GetCameraFromIPAddress

- FlyCapture2Managed::Managed-
BusManager, [76](#)
- GetCameraFromIndex
 - FlyCapture2Managed::Managed-
BusManager, [75](#)
- GetCameraFromSerialNumber
 - FlyCapture2Managed::Managed-
BusManager, [76](#)
- GetCameraInfo
 - FlyCapture2Managed::Managed-
CameraBase, [93](#)
- GetCameraSerialNumberFromIndex
 - FlyCapture2Managed::Managed-
BusManager, [76](#)
- GetChannelStatus
 - FlyCapture2Managed::Managed-
ImageStatistics, [128](#)
- GetChild
 - FlyCapture2Managed::Managed-
TopologyNode, [132](#)
- GetConfiguration
 - FlyCapture2Managed::Managed-
CameraBase, [93](#)
- GetCycleTime
 - FlyCapture2Managed::Managed-
CameraBase, [93](#)
- GetDeviceFromIndex
 - FlyCapture2Managed::Managed-
BusManager, [77](#)
- GetDeviceId
 - FlyCapture2Managed::Managed-
TopologyNode, [133](#)
- GetEmbeddedImageInfo
 - FlyCapture2Managed::Managed-
CameraBase, [94](#)
- GetFormat7Configuration
 - FlyCapture2Managed::Managed-
Camera, [83](#)
- GetFormat7Info
 - FlyCapture2Managed::Managed-
Camera, [84](#)
- GetGPIOPinDirection
 - FlyCapture2Managed::Managed-
CameraBase, [94](#)
- GetGigEConfig
 - FlyCapture2Managed::ManagedGig-
ECamera, [113](#)
- GetGigEImageBinningSettings
 - FlyCapture2Managed::ManagedGig-
ECamera, [114](#)
- GetGigEImageSettings
 - FlyCapture2Managed::ManagedGig-
ECamera, [114](#)
- GetGigEImageSettingsInfo
 - FlyCapture2Managed::ManagedGig-
ECamera, [114](#)
- GetGigEImagingMode
 - FlyCapture2Managed::ManagedGig-
ECamera, [114](#)
- GetGigEProperty
 - FlyCapture2Managed::ManagedGig-
ECamera, [114](#)
- GetGigEStreamChannelInfo
 - FlyCapture2Managed::ManagedGig-
ECamera, [115](#)
- GetGuid
 - FlyCapture2Managed::Managed-
TopologyNode, [133](#)
- GetHashCode
 - FlyCapture2Managed::ManagedPG-
RGuid, [130](#)
- GetHistogram
 - FlyCapture2Managed::Managed-
ImageStatistics, [128](#)
- GetInterfaceType
 - FlyCapture2Managed::Managed-
TopologyNode, [133](#)
- GetInterfaceTypeFromGuid
 - FlyCapture2Managed::Managed-
BusManager, [77](#)
- GetLUTBankInfo
 - FlyCapture2Managed::Managed-
CameraBase, [94](#)
- GetLUTChannel
 - FlyCapture2Managed::Managed-
CameraBase, [95](#)
- GetLUTInfo
 - FlyCapture2Managed::Managed-
CameraBase, [95](#)
- GetMean
 - FlyCapture2Managed::Managed-
ImageStatistics, [128](#)
- GetMemoryChannel
 - FlyCapture2Managed::Managed-
CameraBase, [95](#)
- GetMemoryChannelInfo
 - FlyCapture2Managed::Managed-
CameraBase, [96](#)
- GetNativeCamera

- FlyCapture2Managed::Managed-CameraBase, [96](#)
- GetNativeImage
 - FlyCapture2Managed::Managed-Image, [123](#)
- GetNativeImageStatistics
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- GetNodeMap
 - FlyCapture2Managed::ManagedGC-Camera, [109](#)
- GetNodeType
 - FlyCapture2Managed::Managed-TopologyNode, [133](#)
- GetNumChildren
 - FlyCapture2Managed::Managed-TopologyNode, [133](#)
- GetNumOfCameras
 - FlyCapture2Managed::Managed-BusManager, [77](#)
- GetNumOfDevices
 - FlyCapture2Managed::Managed-BusManager, [77](#)
- GetNumPixelValues
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- GetNumPorts
 - FlyCapture2Managed::Managed-TopologyNode, [133](#)
- GetNumStreamChannels
 - FlyCapture2Managed::ManagedGigE-Camera, [115](#)
- GetPixelValueRange
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- GetPortType
 - FlyCapture2Managed::Managed-TopologyNode, [134](#)
- GetProperty
 - FlyCapture2Managed::Managed-CameraBase, [96](#)
- GetPropertyInfo
 - FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetRange
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- GetRawNativeImagePointer
 - FlyCapture2Managed::Managed-Image, [123](#)
- GetRegisterString
 - FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetSelectedCameraGuids
 - FlyCapture2Managed::Gui::Camera-SelectionDialog, [44](#)
- GetStatistics
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- GetStrobe
 - FlyCapture2Managed::Managed-CameraBase, [97](#)
- GetStrobeInfo
 - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetTopology
 - FlyCapture2Managed::Managed-BusManager, [78](#)
- GetTriggerMode
 - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetTriggerModelInfo
 - FlyCapture2Managed::Managed-CameraBase, [98](#)
- GetUsbLinkInfo
 - FlyCapture2Managed::Managed-BusManager, [78](#)
- GetUsbPortStatus
 - FlyCapture2Managed::Managed-BusManager, [78](#)
- GetVideoModeAndFrameRate
 - FlyCapture2Managed::Managed-Camera, [84](#)
- GetVideoModeAndFrameRateInfo
 - FlyCapture2Managed::Managed-Camera, [84](#)
- GigEConfig, [57](#)
- GigEImageSettings, [58](#)
- GigEImageSettingsInfo, [59](#)
- GigEProperty, [61](#)
- GigEPropertyType
 - Enumerations, [17](#)
- GigEStreamChannel, [62](#)
- GrabMode
 - Enumerations, [17](#)
- GrabTimeout
 - Enumerations, [18](#)
- H264Option, [63](#)
 - FlyCapture2Managed::H264Option, [64](#)

- Hide
 - FlyCapture2Managed::Gui::Camera-
ControlDialog, [34](#)
- Image saving structures., [26](#)
- ImageCallbackDelegate
 - FlyCapture2Managed, [32](#)
- ImageEventCallback
 - FlyCapture2Managed, [32](#)
- ImageFileFormat
 - Enumerations, [18](#)
- ImageMetadata, [64](#)
- InterfaceType
 - Enumerations, [18](#)
- IsCameraControllable
 - FlyCapture2Managed::Managed-
BusManager, [78](#)
- IsConnected
 - FlyCapture2Managed::Managed-
CameraBase, [98](#)
- IsNativeImageValid
 - FlyCapture2Managed::Managed-
Image, [123](#)
- IsVisible
 - FlyCapture2Managed::Gui::Camera-
ControlDialog, [34](#)
- JpegOption, [66](#)
 - FlyCapture2Managed::JpegOption,
[67](#)
- Jpg2Option, [67](#)
 - FlyCapture2Managed::Jpg2Option,
[67](#)
- LaunchBrowser
 - FlyCapture2Managed::Managed-
Utilities, [135](#)
- LaunchCommand
 - FlyCapture2Managed::Managed-
Utilities, [135](#)
- LaunchHelp
 - FlyCapture2Managed::Managed-
Utilities, [135](#)
- LutData, [68](#)
- MJPGOption, [135](#)
 - FlyCapture2Managed::MJPGOption,
[135](#)
- ManagedAVIRecorder, [69](#)
 - FlyCapture2Managed::ManagedAVI-
Recorder, [70](#)
- ManagedBusManager, [71](#)
 - FlyCapture2Managed::Managed-
BusManager, [73](#)
- ManagedCallbackType
 - Enumerations, [19](#)
- ManagedCamera, [80](#)
 - FlyCapture2Managed::Managed-
Camera, [83](#)
- ManagedCameraBase, [88](#)
 - FlyCapture2Managed::Managed-
CameraBase, [92](#)
- ManagedGCCamera, [107](#)
 - FlyCapture2Managed::ManagedGC-
Camera, [108](#)
- ManagedGCPort, [109](#)
 - FlyCapture2Managed::ManagedGC-
Port, [109](#)
- ManagedGigECamera, [109](#)
 - FlyCapture2Managed::ManagedGig-
ECamera, [113](#)
- ManagedImage, [118](#)
 - FlyCapture2Managed::Managed-
Image, [121](#), [122](#)
- ManagedImageStatistics, [127](#)
 - FlyCapture2Managed::Managed-
ImageStatistics, [128](#)
- ManagedPGRGuid, [128](#)
 - FlyCapture2Managed::ManagedPG-
RGuid, [129](#)
- ManagedTopologyNode, [130](#)
 - FlyCapture2Managed::Managed-
TopologyNode, [132](#)
- ManagedUtilities, [134](#)
- Mode
 - Enumerations, [19](#)
- NativeErrorTrace
 - FlyCapture2Managed::FC2Exception,
[52](#)
- NodeType
 - FlyCapture2Managed::Managed-
TopologyNode, [132](#)
- OSType
 - Enumerations, [20](#)
- OnNativeCallback
 - FlyCapture2Managed::Managed-
CameraBase, [99](#)
- PCleBusSpeed
 - Enumerations, [20](#)
- PgmOption, [136](#)
 - FlyCapture2Managed::PgmOption,
[136](#)
- PixelFormat
 - Enumerations, [21](#)

- PngOption, [136](#)
 - FlyCapture2Managed::PngOption, [137](#)
- PortType
 - FlyCapture2Managed::Managed-TopologyNode, [132](#)
- PpmOption, [137](#)
 - FlyCapture2Managed::PpmOption, [138](#)
- PropertyType
 - Enumerations, [21](#)
- QueryGigEImagingMode
 - FlyCapture2Managed::ManagedGig-ECamera, [115](#)
- ROIPosition
 - FlyCapture2Managed::Embedded-ImageInfo, [48](#)
- Read
 - FlyCapture2Managed::ManagedGC-Port, [109](#)
- ReadGVCPMemory
 - FlyCapture2Managed::ManagedGig-ECamera, [115](#)
- ReadGVCPRegister
 - FlyCapture2Managed::ManagedGig-ECamera, [116](#)
- ReadGVCPRegisterBlock
 - FlyCapture2Managed::ManagedGig-ECamera, [116](#)
- ReadPhyRegister
 - FlyCapture2Managed::Managed-BusManager, [79](#)
- ReadRegister
 - FlyCapture2Managed::Managed-CameraBase, [99](#)
- ReadRegisterBlock
 - FlyCapture2Managed::Managed-CameraBase, [99](#)
- RegisterCallback
 - FlyCapture2Managed::Managed-BusManager, [79](#)
- ReleaseBuffer
 - FlyCapture2Managed::Managed-Image, [123](#)
- RescanBus
 - FlyCapture2Managed::Managed-BusManager, [79](#)
- RestoreFromMemoryChannel
 - FlyCapture2Managed::Managed-CameraBase, [99](#)
- RetrieveBuffer
 - FlyCapture2Managed::Managed-CameraBase, [100](#)
- Save
 - FlyCapture2Managed::Managed-Image, [123](#), [124](#)
- SaveToMemoryChannel
 - FlyCapture2Managed::Managed-CameraBase, [100](#)
- SetActiveLUTBank
 - FlyCapture2Managed::Managed-CameraBase, [101](#)
- SetCallback
 - FlyCapture2Managed::Managed-CameraBase, [101](#)
- SetCamera
 - FlyCapture2Managed::Managed-CameraBase, [101](#)
 - FlyCapture2Managed::ManagedGC-Camera, [109](#)
- SetChannelStatus
 - FlyCapture2Managed::Managed-ImageStatistics, [128](#)
- SetConfiguration
 - FlyCapture2Managed::Managed-CameraBase, [101](#)
- SetData
 - FlyCapture2Managed::Managed-Image, [125](#)
- SetDimensions
 - FlyCapture2Managed::Managed-Image, [125](#)
- SetEmbeddedImageInfo
 - FlyCapture2Managed::Managed-CameraBase, [101](#)
- SetFormat7Configuration
 - FlyCapture2Managed::Managed-Camera, [85](#)
- SetGPIOPinDirection
 - FlyCapture2Managed::Managed-CameraBase, [102](#)
- SetGigEConfig
 - FlyCapture2Managed::ManagedGig-ECamera, [116](#)
- SetGigEImageBinningSettings
 - FlyCapture2Managed::ManagedGig-ECamera, [116](#)
- SetGigEImageSettings
 - FlyCapture2Managed::ManagedGig-ECamera, [117](#)

- SetGigEImagingMode
 - FlyCapture2Managed::ManagedGig-ECamera, [117](#)
- SetGigEProperty
 - FlyCapture2Managed::ManagedGig-ECamera, [117](#)
- SetGigEStreamChannelInfo
 - FlyCapture2Managed::ManagedGig-ECamera, [117](#)
- SetLUTChannel
 - FlyCapture2Managed::Managed-CameraBase, [103](#)
- SetProperty
 - FlyCapture2Managed::Managed-CameraBase, [103](#)
- SetStrobe
 - FlyCapture2Managed::Managed-CameraBase, [104](#)
- SetTitle
 - FlyCapture2Managed::Gui::Camera-ControlDialog, [34](#)
 - FlyCapture2Managed::Gui::Camera-SelectionDialog, [44](#)
- SetTriggerMode
 - FlyCapture2Managed::Managed-CameraBase, [104](#)
- SetVideoModeAndFrameRate
 - FlyCapture2Managed::Managed-Camera, [86](#)
- Show
 - FlyCapture2Managed::Gui::Camera-ControlDialog, [34](#)
- ShowModal
 - FlyCapture2Managed::Gui::Camera-SelectionDialog, [45](#)
- StartCapture
 - FlyCapture2Managed::Managed-CameraBase, [104](#), [105](#)
- StartSyncCapture
 - FlyCapture2Managed::Managed-Camera, [86](#)
- StatisticsChannel
 - Enumerations, [22](#)
- StopCapture
 - FlyCapture2Managed::Managed-CameraBase, [105](#)
- StrobeControl, [138](#)
- StrobeInfo, [139](#)
- Structures, [24](#)
- SystemInfo, [140](#)
- TiffOption, [142](#)
 - FlyCapture2Managed::TiffOption, [143](#)
- TimeStamp, [143](#)
- ToMgd
 - FlyCapture2Managed::Translate, [147–149](#)
- ToNative
 - FlyCapture2Managed::Translate, [149–151](#)
- Translate, [144](#)
- Translate::ToMgd
 - FlyCapture2Managed::Translate, [154](#)
- Translate::ToNative
 - FlyCapture2Managed::Translate, [154](#)
- TranslateNodeType
 - FlyCapture2Managed::Managed-TopologyNode, [134](#)
- TranslatePortType
 - FlyCapture2Managed::Managed-TopologyNode, [134](#)
- TriggerMode, [154](#)
- TriggerModelInfo, [155](#)
- Type
 - FlyCapture2Managed::FC2Exception, [52](#)
- UnregisterCallback
 - FlyCapture2Managed::Managed-BusManager, [80](#)
- ValidateFormat7Settings
 - FlyCapture2Managed::Managed-Camera, [87](#)
- VideoMode
 - Enumerations, [22](#)
- WaitForBufferEvent
 - FlyCapture2Managed::Managed-CameraBase, [105](#)
- Write
 - FlyCapture2Managed::ManagedGC-Port, [109](#)
- WriteGVCPMemory
 - FlyCapture2Managed::ManagedGig-ECamera, [117](#)
- WriteGVCPRegister
 - FlyCapture2Managed::ManagedGig-ECamera, [118](#)
- WriteGVCPRegisterBlock
 - FlyCapture2Managed::ManagedGig-ECamera, [118](#)
- WritePhyRegister

- FlyCapture2Managed::Managed-BusManager, 80
- WriteRegister
 - FlyCapture2Managed::Managed-CameraBase, 105, 106
- WriteRegisterBlock
 - FlyCapture2Managed::Managed-CameraBase, 106
- absControl
 - FlyCapture2Managed::Camera-Property, 40
- absMax
 - FlyCapture2Managed::Camera-PropertyInfo, 42
- absMin
 - FlyCapture2Managed::Camera-PropertyInfo, 42
- absValSupported
 - FlyCapture2Managed::Camera-PropertyInfo, 42
- absValue
 - FlyCapture2Managed::Camera-Property, 40
- applicationIPAddress
 - FlyCapture2Managed::CameraInfo, 36
- applicationPort
 - FlyCapture2Managed::CameraInfo, 36
- asyncBusSpeed
 - FlyCapture2Managed::FC2Config, 50
- autoManualMode
 - FlyCapture2Managed::Camera-Property, 40
- autoSupported
 - FlyCapture2Managed::Camera-PropertyInfo, 43
- available
 - FlyCapture2Managed::Embedded-ImageInfoProperty, 48
- bandwidthAllocation
 - FlyCapture2Managed::FC2Config, 50
- bayerTileFormat
 - FlyCapture2Managed::CameraInfo, 36
- FlyCapture2Managed::Managed-Image, 125
- binaryFile
 - FlyCapture2Managed::PgmOption, 136
 - FlyCapture2Managed::PpmOption, 138
- bitmap
 - FlyCapture2Managed::Managed-Image, 125
- bitrate
 - FlyCapture2Managed::H264Option, 64
- bitsPerPixel
 - FlyCapture2Managed::Managed-Image, 125
- brightness
 - FlyCapture2Managed::Embedded-ImageInfo, 47
- build
 - FlyCapture2Managed::FC2Version, 53
- busNumber
 - FlyCapture2Managed::CameraInfo, 36
- byteOrder
 - FlyCapture2Managed::SystemInfo, 141
- ccpStatus
 - FlyCapture2Managed::CameraInfo, 37
- chipIdHi
 - FlyCapture2Managed::ConfigROM, 46
- chipIdLo
 - FlyCapture2Managed::ConfigROM, 46
- colorProcessingAlgorithm
 - FlyCapture2Managed::Managed-Image, 126
- cols
 - FlyCapture2Managed::Managed-Image, 126
- compression
 - FlyCapture2Managed::TiffOption, 143
- compressionLevel
 - FlyCapture2Managed::PngOption, 137

- configROM
 - FlyCapture2Managed::CameraInfo, 37
- cpuDescription
 - FlyCapture2Managed::SystemInfo, 141
- cycleCount
 - FlyCapture2Managed::TimeStamp, 144
- cycleOffset
 - FlyCapture2Managed::TimeStamp, 144
- cycleSeconds
 - FlyCapture2Managed::TimeStamp, 144
- data
 - FlyCapture2Managed::Managed-Image, 126
- defaultColorProcessingAlgorithm
 - FlyCapture2Managed::Managed-Image, 126
- defaultGateway
 - FlyCapture2Managed::CameraInfo, 37
- defaultOutputPixelFormat
 - FlyCapture2Managed::Managed-Image, 126
- delay
 - FlyCapture2Managed::Strobe-Control, 138
- destinationIpAddress
 - FlyCapture2Managed::GigEStream-Channel, 62
- doNotFragment
 - FlyCapture2Managed::GigEStream-Channel, 62
- driverList
 - FlyCapture2Managed::SystemInfo, 141
- driverName
 - FlyCapture2Managed::CameraInfo, 37
- driverType
 - FlyCapture2Managed::CameraInfo, 37
- duration
 - FlyCapture2Managed::Strobe-Control, 138
- embeddedBrightness
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedExposure
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedFrameCounter
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedGPIOPinState
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedGain
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedROIPosition
 - FlyCapture2Managed::Image-Metadata, 65
- embeddedShutter
 - FlyCapture2Managed::Image-Metadata, 66
- embeddedStrobePattern
 - FlyCapture2Managed::Image-Metadata, 66
- embeddedTimeStamp
 - FlyCapture2Managed::Image-Metadata, 66
- embeddedWhiteBalance
 - FlyCapture2Managed::Image-Metadata, 66
- enablePacketResend
 - FlyCapture2Managed::GigEConfig, 58
- enabled
 - FlyCapture2Managed::LutData, 68
- exposure
 - FlyCapture2Managed::Embedded-ImageInfo, 47
- firmwareBuildTime
 - FlyCapture2Managed::CameraInfo, 37
- firmwareVersion
 - FlyCapture2Managed::CameraInfo, 37
- frameCounter
 - FlyCapture2Managed::Embedded-ImageInfo, 47
- frameRate
 - FlyCapture2Managed::AviOption, 33

- FlyCapture2Managed::H264Option, [64](#)
 - FlyCapture2Managed::MJPGOption, [135](#)
- gain
 - FlyCapture2Managed::Embedded-ImageInfo, [48](#)
- gigEMajorVersion
 - FlyCapture2Managed::CameraInfo, [37](#)
- gigEMinorVersion
 - FlyCapture2Managed::CameraInfo, [37](#)
- gpuDescription
 - FlyCapture2Managed::SystemInfo, [141](#)
- grabMode
 - FlyCapture2Managed::FC2Config, [50](#)
- grabTimeout
 - FlyCapture2Managed::FC2Config, [50](#)
- height
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::GigEImageSettings, [59](#)
 - FlyCapture2Managed::H264Option, [64](#)
- hostPost
 - FlyCapture2Managed::GigEStream-Channel, [63](#)
- iidcVersion
 - FlyCapture2Managed::CameraInfo, [37](#)
- imageHStepSize
 - FlyCapture2Managed::Format7Info, [55](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [60](#)
- imageMetadata
 - FlyCapture2Managed::Managed-Image, [126](#)
- imageVStepSize
 - FlyCapture2Managed::Format7Info, [55](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [60](#)
- inputBitDepth
 - FlyCapture2Managed::LutData, [68](#)
- interPacketDelay
 - FlyCapture2Managed::GigEStream-Channel, [63](#)
- interfaceType
 - FlyCapture2Managed::CameraInfo, [38](#)
- interlaced
 - FlyCapture2Managed::PngOption, [137](#)
- ipAddress
 - FlyCapture2Managed::CameraInfo, [38](#)
- isColorCamera
 - FlyCapture2Managed::CameraInfo, [38](#)
- isReadable
 - FlyCapture2Managed::GigEProperty, [61](#)
- isWritable
 - FlyCapture2Managed::GigEProperty, [61](#)
- isochBusSpeed
 - FlyCapture2Managed::FC2Config, [50](#)
- keyword
 - FlyCapture2Managed::ConfigROM, [46](#)
- libraryList
 - FlyCapture2Managed::SystemInfo, [141](#)
- libraryVersion
 - FlyCapture2Managed::Managed-Utilities, [135](#)
- m_externalDelegate
 - FlyCapture2Managed::Managed-CameraBase, [106](#)
- m_internalDelegate
 - FlyCapture2Managed::Managed-CameraBase, [106](#)
- m_isLocal
 - FlyCapture2Managed::Managed-CameraBase, [107](#)
- m_pNativeCamBase
 - FlyCapture2Managed::Managed-CameraBase, [107](#)

- macAddress
 - FlyCapture2Managed::CameraInfo, [38](#)
- major
 - FlyCapture2Managed::FC2Version, [53](#)
- manualSupported
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
- max
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
 - FlyCapture2Managed::GigEProperty, [61](#)
- maxBytesPerPacket
 - FlyCapture2Managed::Format7-PacketInfo, [57](#)
- maxHeight
 - FlyCapture2Managed::Format7Info, [55](#)
 - FlyCapture2Managed::GigEImage-SettingsInfo, [60](#)
- maxPacketSize
 - FlyCapture2Managed::Format7Info, [56](#)
- maxValue
 - FlyCapture2Managed::StrobeInfo, [139](#)
- maxWidth
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEImage-SettingsInfo, [60](#)
- maximumBusSpeed
 - FlyCapture2Managed::CameraInfo, [38](#)
- microSeconds
 - FlyCapture2Managed::TimeStamp, [144](#)
- min
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
 - FlyCapture2Managed::GigEProperty, [61](#)
- minNumImageNotifications
 - FlyCapture2Managed::FC2Config, [50](#)
- minPacketSize
 - FlyCapture2Managed::Format7Info, [56](#)
- minValue
 - FlyCapture2Managed::StrobeInfo, [139](#)
- minor
 - FlyCapture2Managed::FC2Version, [53](#)
- mode
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::TriggerMode, [154](#)
- modeMask
 - FlyCapture2Managed::TriggerMode-Info, [156](#)
- modelName
 - FlyCapture2Managed::CameraInfo, [38](#)
- networkInterfaceIndex
 - FlyCapture2Managed::GigEStream-Channel, [63](#)
- nodeNumber
 - FlyCapture2Managed::CameraInfo, [38](#)
- nodeVendorId
 - FlyCapture2Managed::ConfigROM, [46](#)
- numBanks
 - FlyCapture2Managed::LutData, [68](#)
- numBuffers
 - FlyCapture2Managed::FC2Config, [50](#)
- numChannels
 - FlyCapture2Managed::LutData, [69](#)
- numCpuCores
 - FlyCapture2Managed::SystemInfo, [141](#)
- numEntries
 - FlyCapture2Managed::LutData, [69](#)
- numImageNotifications
 - FlyCapture2Managed::FC2Config, [50](#)
- offsetHStepSize
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEImage-SettingsInfo, [60](#)

- offsetVStepSize
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [60](#)
- offsetX
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::GigEImageSettings, [59](#)
- offsetY
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::GigEImageSettings, [59](#)
- onOff
 - FlyCapture2Managed::Camera-Property, [41](#)
 - FlyCapture2Managed::Embedded-ImageInfoProperty, [48](#)
 - FlyCapture2Managed::Strobe-Control, [138](#)
 - FlyCapture2Managed::TriggerMode, [154](#)
- onOffSupported
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
 - FlyCapture2Managed::StrobeInfo, [140](#)
 - FlyCapture2Managed::TriggerMode-Info, [156](#)
- onePush
 - FlyCapture2Managed::Camera-Property, [40](#)
- onePushSupported
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
- operator=
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- operator==
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- osDescription
 - FlyCapture2Managed::SystemInfo, [141](#)
- osType
 - FlyCapture2Managed::SystemInfo, [142](#)
- outputBitDepth
 - FlyCapture2Managed::LutData, [69](#)
- packetSize
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEStream-Channel, [63](#)
- parameter
 - FlyCapture2Managed::TriggerMode, [154](#)
- pcieBusSpeed
 - FlyCapture2Managed::CameraInfo, [38](#)
- percentage
 - FlyCapture2Managed::Format7Info, [56](#)
- pixelFormat
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::GigEImageSettings, [59](#)
 - FlyCapture2Managed::Managed-Image, [126](#)
- pixelFormatBitField
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEImageSettingsInfo, [60](#)
- polarity
 - FlyCapture2Managed::Strobe-Control, [139](#)
 - FlyCapture2Managed::TriggerMode, [155](#)
- polaritySupported
 - FlyCapture2Managed::StrobeInfo, [140](#)
 - FlyCapture2Managed::TriggerMode-Info, [156](#)
- present
 - FlyCapture2Managed::Camera-Property, [41](#)
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
 - FlyCapture2Managed::StrobeInfo, [140](#)
 - FlyCapture2Managed::TriggerMode-Info, [156](#)
- progressive
 - FlyCapture2Managed::JpegOption, [67](#)

- propType
 - FlyCapture2Managed::GigEProperty, 61
- quality
 - FlyCapture2Managed::JpegOption, 67
 - FlyCapture2Managed::Jpg2Option, 68
 - FlyCapture2Managed::MJPGOption, 135
- readOutSupported
 - FlyCapture2Managed::CameraPropertyInfo, 43
 - FlyCapture2Managed::StrobeInfo, 140
 - FlyCapture2Managed::TriggerModeInfo, 156
- receivedDataSize
 - FlyCapture2Managed::ManagedImage, 126
- recommendedBytesPerPacket
 - FlyCapture2Managed::Format7PacketInfo, 57
- registerTimeout
 - FlyCapture2Managed::FC2Config, 51
- registerTimeoutRetries
 - FlyCapture2Managed::FC2Config, 51
- rows
 - FlyCapture2Managed::ManagedImage, 126
- screenHeight
 - FlyCapture2Managed::SystemInfo, 142
- screenWidth
 - FlyCapture2Managed::SystemInfo, 142
- seconds
 - FlyCapture2Managed::TimeStamp, 144
- sensorInfo
 - FlyCapture2Managed::CameraInfo, 38
- sensorResolution
 - FlyCapture2Managed::CameraInfo, 38
- serialNumber
 - FlyCapture2Managed::CameraInfo, 39
- shutter
 - FlyCapture2Managed::EmbeddedImageInfo, 48
- softwareTriggerSupported
 - FlyCapture2Managed::TriggerModeInfo, 156
- source
 - FlyCapture2Managed::StrobeControl, 139
 - FlyCapture2Managed::StrobeInfo, 140
 - FlyCapture2Managed::TriggerMode, 155
- sourceMask
 - FlyCapture2Managed::TriggerModeInfo, 156
- sourcePort
 - FlyCapture2Managed::GigEStreamChannel, 63
- stride
 - FlyCapture2Managed::ManagedImage, 127
- strobePattern
 - FlyCapture2Managed::EmbeddedImageInfo, 48
- subnetMask
 - FlyCapture2Managed::CameraInfo, 39
- supported
 - FlyCapture2Managed::LutData, 69
- systemInfo
 - FlyCapture2Managed::ManagedUtilities, 135
- systemMemorySize
 - FlyCapture2Managed::SystemInfo, 142
- timeStamp
 - FlyCapture2Managed::ManagedImage, 127
- timestamp
 - FlyCapture2Managed::EmbeddedImageInfo, 48
- translate
 - FlyCapture2Managed::Translate, 151–154
- type

- FlyCapture2Managed::Camera-Property, [41](#)
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
 - FlyCapture2Managed::FC2Version, [53](#)
- unitAbbr
 - FlyCapture2Managed::Camera-PropertyInfo, [43](#)
- unitBytesPerPacket
 - FlyCapture2Managed::Format7-PacketInfo, [57](#)
- unitSWVer
 - FlyCapture2Managed::ConfigROM, [46](#)
- unitSpecId
 - FlyCapture2Managed::ConfigROM, [46](#)
- unitSubSWVer
 - FlyCapture2Managed::ConfigROM, [46](#)
- units
 - FlyCapture2Managed::Camera-PropertyInfo, [44](#)
- userDefinedName
 - FlyCapture2Managed::CameraInfo, [39](#)
- value
 - FlyCapture2Managed::GigEProperty, [62](#)
- value0
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- value1
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- value2
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- value3
 - FlyCapture2Managed::ManagedPG-RGuid, [130](#)
- valueA
 - FlyCapture2Managed::Camera-Property, [41](#)
- valueB
 - FlyCapture2Managed::Camera-Property, [41](#)
- valueReadable
 - FlyCapture2Managed::TriggerMode-Info, [156](#)
- vendorName
 - FlyCapture2Managed::CameraInfo, [39](#)
- vendorPixelFormatBitField
 - FlyCapture2Managed::Format7Info, [56](#)
 - FlyCapture2Managed::GigEImage-SettingsInfo, [60](#)
- vendorUniqueInfo0
 - FlyCapture2Managed::ConfigROM, [46](#)
- vendorUniqueInfo1
 - FlyCapture2Managed::ConfigROM, [46](#)
- vendorUniqueInfo2
 - FlyCapture2Managed::ConfigROM, [46](#)
- vendorUniqueInfo3
 - FlyCapture2Managed::ConfigROM, [47](#)
- whiteBalance
 - FlyCapture2Managed::Embedded-ImageInfo, [48](#)
- width
 - FlyCapture2Managed::Format7-ImageSettings, [54](#)
 - FlyCapture2Managed::GigEImage-Settings, [59](#)
 - FlyCapture2Managed::H264Option, [64](#)
- xmlURL1
 - FlyCapture2Managed::CameraInfo, [39](#)
- xmlURL2
 - FlyCapture2Managed::CameraInfo, [39](#)