FlyCapture2 C

2.7.3.17

Generated by Doxygen 1.7.5

Thu Feb 5 2015 09:31:47

Contents

1 Deprecated List **1**

2 Data Structure Index **3**

2.1 Data Structures 3

3 File Index **5**

3.1 File List 5

4 Data Structure Documentation **7**

4.1 fc2AVIOption Struct Reference 7

4.1.1 Field Documentation 7

4.1.1.1 frameRate 7

4.1.1.2 reserved 7

4.2 fc2CameraInfo Struct Reference 8

4.2.1 Field Documentation 9

4.2.1.1 applicationIPAddress 9

4.2.1.2 applicationPort 9

4.2.1.3 bayerTileFormat 9

4.2.1.4 busNumber 9

4.2.1.5 ccpStatus 9

4.2.1.6 configROM 9

4.2.1.7 defaultGateway 9

4.2.1.8 driverName 9

4.2.1.9 driverType 9

4.2.1.10 firmwareBuildTime 9

4.2.1.11 firmwareVersion 9

4.2.1.12 gigEMajorVersion 9
4.2.1.13 gigEMinorVersion 10
4.2.1.14 iidcVer 10
4.2.1.15 interfaceType 10
4.2.1.16 ipAddress 10
4.2.1.17 isColorCamera 10
4.2.1.18 macAddress 10
4.2.1.19 maximumBusSpeed 10
4.2.1.20 modelName 10
4.2.1.21 nodeNumber 10
4.2.1.22 pcieBusSpeed 10
4.2.1.23 reserved 10
4.2.1.24 sensorInfo 10
4.2.1.25 sensorResolution 10
4.2.1.26 serialNumber 10
4.2.1.27 subnetMask 10
4.2.1.28 userDefinedName 10
4.2.1.29 vendorName 10
4.2.1.30 xmlURL1 10
4.2.1.31 xmlURL2 10
4.3 fc2Config Struct Reference 10
4.3.1 Field Documentation 11
4.3.1.1 asyncBusSpeed 11
4.3.1.2 bandwidthAllocation 11
4.3.1.3 grabMode 11
4.3.1.4 grabTimeout 11
4.3.1.5 isochBusSpeed 11
4.3.1.6 minNumImageNotifications 11
4.3.1.7 numBuffers 11
4.3.1.8 numImageNotifications 11
4.3.1.9 registerTimeout 11
4.3.1.10 registerTimeoutRetries 11
4.3.1.11 reserved 11
4.4 fc2ConfigROM Struct Reference 11

4.4.1 Field Documentation 12
4.4.1.1 chipIdHi 12
4.4.1.2 chipIdLo 12
4.4.1.3 nodeVendorId 12
4.4.1.4 pszKeyword 12
4.4.1.5 reserved 12
4.4.1.6 unitSpecId 12
4.4.1.7 unitSubSWVer 12
4.4.1.8 unitSWVer 12
4.4.1.9 vendorUniqueInfo_0 12
4.4.1.10 vendorUniqueInfo_1 12
4.4.1.11 vendorUniqueInfo_2 12
4.4.1.12 vendorUniqueInfo_3 12
4.5 fc2EmbeddedImageInfo Struct Reference 13
4.5.1 Field Documentation 13
4.5.1.1 brightness 13
4.5.1.2 exposure 14
4.5.1.3 frameCounter 14
4.5.1.4 gain 14
4.5.1.5 GPIOPinState 14
4.5.1.6 ROIPosition 14
4.5.1.7 shutter 14
4.5.1.8 strobePattern 14
4.5.1.9 timestamp 14
4.5.1.10 whiteBalance 14
4.6 fc2EmbeddedImageInfoProperty Struct Reference 14
4.6.1 Field Documentation 14
4.6.1.1 available 14
4.6.1.2 onOff 14
4.7 fc2Format7ImageSettings Struct Reference 14
4.7.1 Field Documentation 15
4.7.1.1 height 15
4.7.1.2 mode 15
4.7.1.3 offsetX 15

4.7.1.4 offsetY 15
4.7.1.5 pixelFormat 15
4.7.1.6 reserved 15
4.7.1.7 width 15
4.8 fc2Format7Info Struct Reference 15
4.8.1 Field Documentation 16
4.8.1.1 imageHStepSize 16
4.8.1.2 imageVStepSize 16
4.8.1.3 maxHeight 16
4.8.1.4 maxPacketSize 16
4.8.1.5 maxWidth 16
4.8.1.6 minPacketSize 16
4.8.1.7 mode 16
4.8.1.8 offsetHStepSize 16
4.8.1.9 offsetVStepSize 16
4.8.1.10 packetSize 16
4.8.1.11 percentage 16
4.8.1.12 pixelFormatBitField 16
4.8.1.13 reserved 16
4.8.1.14 vendorPixelFormatBitField 16
4.9 fc2Format7PacketInfo Struct Reference 16
4.9.1 Field Documentation 16
4.9.1.1 maxBytesPerPacket 17
4.9.1.2 recommendedBytesPerPacket 17
4.9.1.3 reserved 17
4.9.1.4 unitBytesPerPacket 17
4.10 fc2GigEConfig Struct Reference 17
4.10.1 Field Documentation 17
4.10.1.1 enablePacketResend 17
4.10.1.2 maxPacketsToResend 17
4.10.1.3 reserved 17
4.10.1.4 timeoutForPacketResend 17
4.11 fc2GigEImageSettings Struct Reference 18
4.11.1 Field Documentation 18

4.11.1.1 height 18
4.11.1.2 offsetX 18
4.11.1.3 offsetY 18
4.11.1.4 pixelFormat 18
4.11.1.5 reserved 18
4.11.1.6 width 18
4.12 fc2GigEImageSettingsInfo Struct Reference 18
4.12.1 Field Documentation 19
4.12.1.1 imageHStepSize 19
4.12.1.2 imageVStepSize 19
4.12.1.3 maxHeight 19
4.12.1.4 maxWidth 19
4.12.1.5 offsetHStepSize 19
4.12.1.6 offsetVStepSize 19
4.12.1.7 pixelFormatBitField 19
4.12.1.8 reserved 19
4.12.1.9 vendorPixelFormatBitField 19
4.13 fc2GigEProperty Struct Reference 19
4.13.1 Field Documentation 19
4.13.1.1 isReadable 19
4.13.1.2 isWritable 19
4.13.1.3 max 19
4.13.1.4 min 19
4.13.1.5 propType 20
4.13.1.6 reserved 20
4.13.1.7 value 20
4.14 fc2GigEStreamChannel Struct Reference 20
4.14.1 Field Documentation 20
4.14.1.1 destinationIpAddress 20
4.14.1.2 doNotFragment 21
4.14.1.3 hostPost 21
4.14.1.4 interPacketDelay 21
4.14.1.5 networkInterfaceIndex 21
4.14.1.6 packetSize 21

4.14.1.7 reserved 21
4.14.1.8 sourcePort 21
4.15 fc2H264Option Struct Reference 21
4.15.1 Field Documentation 21
4.15.1.1 bitrate 21
4.15.1.2 frameRate 21
4.15.1.3 height 21
4.15.1.4 reserved 21
4.15.1.5 width 21
4.16 fc2Image Struct Reference 22
4.16.1 Field Documentation 22
4.16.1.1 bayerFormat 22
4.16.1.2 cols 22
4.16.1.3 dataSize 22
4.16.1.4 format 22
4.16.1.5 imageImpl 22
4.16.1.6 pData 22
4.16.1.7 receivedDataSize 22
4.16.1.8 rows 22
4.16.1.9 stride 22
4.17 fc2ImageMetadata Struct Reference 22
4.17.1 Field Documentation 23
4.17.1.1 embeddedBrightness 23
4.17.1.2 embeddedExposure 23
4.17.1.3 embeddedFrameCounter 23
4.17.1.4 embeddedGain 23
4.17.1.5 embeddedGPIOPinState 23
4.17.1.6 embeddedROIPosition 23
4.17.1.7 embeddedShutter 23
4.17.1.8 embeddedStrobePattern 23
4.17.1.9 embeddedTimeStamp 23
4.17.1.10 embeddedWhiteBalance 23
4.17.1.11 reserved 23
4.18 fc2InternalContext Struct Reference 23

4.18.1 Field Documentation . . . 24
4.18.1.1 pBusMgr . . . 24
4.18.1.2 pCamera . . . 24
4.19 fc2InternalGuiContext Struct Reference . . . 24
4.19.1 Field Documentation . . . 24
4.19.1.1 pCameraControlDlg . . . 24
4.19.1.2 pCameraSelectionDlg . . . 24
4.20 fc2InternalImageCallback Struct Reference . . . 25
4.20.1 Field Documentation . . . 25
4.20.1.1 pCallback . . . 25
4.20.1.2 pCallbackData . . . 25
4.21 fc2IPAddress Struct Reference . . . 25
4.21.1 Field Documentation . . . 25
4.21.1.1 octets . . . 26
4.22 fc2JPEGOption Struct Reference . . . 26
4.22.1 Field Documentation . . . 26
4.22.1.1 progressive . . . 26
4.22.1.2 quality . . . 26
4.22.1.3 reserved . . . 26
4.23 fc2JPG2Option Struct Reference . . . 26
4.23.1 Field Documentation . . . 26
4.23.1.1 quality . . . 26
4.23.1.2 reserved . . . 26
4.24 fc2LUTData Struct Reference . . . 27
4.24.1 Field Documentation . . . 27
4.24.1.1 enabled . . . 27
4.24.1.2 inputBitDepth . . . 27
4.24.1.3 numBanks . . . 27
4.24.1.4 numChannels . . . 27
4.24.1.5 numEntries . . . 27
4.24.1.6 outputBitDepth . . . 27
4.24.1.7 reserved . . . 27
4.24.1.8 supported . . . 27
4.25 fc2MACAddress Struct Reference . . . 27

4.25.1 Field Documentation 27
4.25.1.1 octets 28
4.26 fc2MJPGOption Struct Reference 28
4.26.1 Field Documentation 28
4.26.1.1 frameRate 28
4.26.1.2 quality 28
4.26.1.3 reserved 28
4.27 fc2PGMOption Struct Reference 28
4.27.1 Field Documentation 28
4.27.1.1 binaryFile 28
4.27.1.2 reserved 28
4.28 fc2PGRGuid Struct Reference 29
4.28.1 Detailed Description 29
4.28.2 Field Documentation 29
4.28.2.1 value 29
4.29 fc2PNGOption Struct Reference 29
4.29.1 Field Documentation 29
4.29.1.1 compressionLevel 29
4.29.1.2 interlaced 29
4.29.1.3 reserved 29
4.30 fc2PPMOption Struct Reference 30
4.30.1 Field Documentation 30
4.30.1.1 binaryFile 30
4.30.1.2 reserved 30
4.31 fc2StrobeControl Struct Reference 30
4.31.1 Field Documentation 30
4.31.1.1 delay 30
4.31.1.2 duration 30
4.31.1.3 onOff 30
4.31.1.4 polarity 30
4.31.1.5 reserved 30
4.31.1.6 source 30
4.32 fc2StrobeInfo Struct Reference 31
4.32.1 Field Documentation 31

4.32.1.1 maxValue 31
4.32.1.2 minValue 31
4.32.1.3 onOffSupported 31
4.32.1.4 polaritySupported 31
4.32.1.5 present 31
4.32.1.6 readOutSupported 31
4.32.1.7 reserved 31
4.32.1.8 source 31
4.33 fc2SystemInfo Struct Reference 31
4.33.1 Field Documentation 32
4.33.1.1 byteOrder 32
4.33.1.2 cpuDescription 32
4.33.1.3 driverList 32
4.33.1.4 gpuDescription 32
4.33.1.5 libraryList 32
4.33.1.6 numCpuCores 32
4.33.1.7 osDescription 32
4.33.1.8 osType 32
4.33.1.9 reserved 32
4.33.1.10 screenHeight 32
4.33.1.11 screenWidth 32
4.33.1.12 sysMemSize 32
4.34 fc2TIFFOption Struct Reference 32
4.34.1 Field Documentation 33
4.34.1.1 compression 33
4.34.1.2 reserved 33
4.35 fc2TimeStamp Struct Reference 33
4.35.1 Field Documentation 33
4.35.1.1 cycleCount 33
4.35.1.2 cycleOffset 33
4.35.1.3 cycleSeconds 33
4.35.1.4 microSeconds 33
4.35.1.5 reserved 33
4.35.1.6 seconds 33

4.36 fc2TriggerDelay Struct Reference 33
4.36.1 Field Documentation 34
4.36.1.1 absControl 34
4.36.1.2 absValue 34
4.36.1.3 autoManualMode 34
4.36.1.4 onePush 34
4.36.1.5 onOff 34
4.36.1.6 present 34
4.36.1.7 reserved 34
4.36.1.8 type 34
4.36.1.9 valueA 34
4.36.1.10 valueB 34
4.37 fc2TriggerDelayInfo Struct Reference 34
4.37.1 Field Documentation 35
4.37.1.1 absMax 35
4.37.1.2 absMin 35
4.37.1.3 absValSupported 35
4.37.1.4 autoSupported 35
4.37.1.5 manualSupported 35
4.37.1.6 max 35
4.37.1.7 min 35
4.37.1.8 onePushSupported 35
4.37.1.9 onOffSupported 35
4.37.1.10 present 35
4.37.1.11 pUnitAbbr 35
4.37.1.12 pUnits 35
4.37.1.13 readOutSupported 35
4.37.1.14 reserved 35
4.37.1.15 type 35
4.38 fc2TriggerMode Struct Reference 36
4.38.1 Field Documentation 36
4.38.1.1 mode 36
4.38.1.2 onOff 36
4.38.1.3 parameter 36

4.38.1.4 polarity 36
4.38.1.5 reserved 36
4.38.1.6 source 36
4.39 fc2TriggerModeInfo Struct Reference 36
4.39.1 Field Documentation 37
4.39.1.1 modeMask 37
4.39.1.2 onOffSupported 37
4.39.1.3 polaritySupported 37
4.39.1.4 present 37
4.39.1.5 readOutSupported 37
4.39.1.6 reserved 37
4.39.1.7 softwareTriggerSupported 37
4.39.1.8 sourceMask 37
4.39.1.9 valueReadable 37
4.40 fc2Version Struct Reference 37
4.40.1 Field Documentation 37
4.40.1.1 build 37
4.40.1.2 major 37
4.40.1.3 minor 37
4.40.1.4 type 37

5 File Documentation 39
5.1 FlyCapture2_C.h File Reference 39
5.1.1 Function Documentation 48
5.1.1.1 fc2AVIAppend 48
5.1.1.2 fc2AVIClose 48
5.1.1.3 fc2AVIOpen 49
5.1.1.4 fc2CalculateImageStatistics 49
5.1.1.5 fc2Connect 50
5.1.1.6 fc2ConvertImage 50
5.1.1.7 fc2ConvertImageTo 50
5.1.1.8 fc2CreateAVI 51
5.1.1.9 fc2CreateContext 51
5.1.1.10 fc2CreateGigEContext 51

5.1.1.11 fc2CreateImage 51
5.1.1.12 fc2CreateImageStatistics 52
5.1.1.13 fc2DestroyAVI 52
5.1.1.14 fc2DestroyContext 52
5.1.1.15 fc2DestroyImage 53
5.1.1.16 fc2DestroyImageStatistics 53
5.1.1.17 fc2DetermineBitsPerPixel 53
5.1.1.18 fc2Disconnect 54
5.1.1.19 fc2DiscoverGigECameras 54
5.1.1.20 fc2EnableLUT 54
5.1.1.21 fc2ErrorToDescription 55
5.1.1.22 fc2FireBusReset 55
5.1.1.23 fc2FireSoftwareTrigger 55
5.1.1.24 fc2FireSoftwareTriggerBroadcast 56
5.1.1.25 fc2ForceAllIPAddressesAutomatically 56
5.1.1.26 fc2ForceIPAddressAutomatically 56
5.1.1.27 fc2ForceIPAddressToCamera 56
5.1.1.28 fc2GetActiveLUTBank 57
5.1.1.29 fc2GetCameraFromIndex 57
5.1.1.30 fc2GetCameraFromIPAddress 58
5.1.1.31 fc2GetCameraFromSerialNumber 58
5.1.1.32 fc2GetCameraInfo 58
5.1.1.33 fc2GetCameraSerialNumberFromIndex 59
5.1.1.34 fc2GetChannelStatus 59
5.1.1.35 fc2GetConfiguration 59
5.1.1.36 fc2GetCycleTime 60
5.1.1.37 fc2GetDefaultColorProcessing 60
5.1.1.38 fc2GetDefaultOutputFormat 60
5.1.1.39 fc2GetDeviceFromIndex 61
5.1.1.40 fc2GetEmbeddedImageInfo 61
5.1.1.41 fc2GetFormat7Configuration 61
5.1.1.42 fc2GetFormat7Info 62
5.1.1.43 fc2GetGigEConfig 62
5.1.1.44 fc2GetGigEImageBinningSettings 62

5.1.1.45 fc2GetGigEImageSettings 62
5.1.1.46 fc2GetGigEImageSettingsInfo 62
5.1.1.47 fc2GetGigEImagingMode 62
5.1.1.48 fc2GetGigEProperty 63
5.1.1.49 fc2GetGigEStreamChannelInfo 63
5.1.1.50 fc2GetGPIOPinDirection 63
5.1.1.51 fc2GetImageData 63
5.1.1.52 fc2GetImageStatistics 64
5.1.1.53 fc2GetImageTimeStamp 64
5.1.1.54 fc2GetInterfaceTypeFromGuid 65
5.1.1.55 fc2GetLibraryVersion 65
5.1.1.56 fc2GetLUTBankInfo 65
5.1.1.57 fc2GetLUTChannel 66
5.1.1.58 fc2GetLUTInfo 66
5.1.1.59 fc2GetMemoryChannel 66
5.1.1.60 fc2GetMemoryChannelInfo 67
5.1.1.61 fc2GetNumOfCameras 67
5.1.1.62 fc2GetNumOfDevices 67
5.1.1.63 fc2GetNumStreamChannels 68
5.1.1.64 fc2GetProperty 68
5.1.1.65 fc2GetPropertyInfo 68
5.1.1.66 fc2GetRegisterString 68
5.1.1.67 fc2GetStrobe 69
5.1.1.68 fc2GetStrobeInfo 69
5.1.1.69 fc2GetSystemInfo 69
5.1.1.70 fc2GetTriggerDelay 70
5.1.1.71 fc2GetTriggerDelayInfo 70
5.1.1.72 fc2GetTriggerMode 70
5.1.1.73 fc2GetTriggerModeInfo 71
5.1.1.74 fc2GetVideoModeAndFrameRate 71
5.1.1.75 fc2GetVideoModeAndFrameRateInfo 71
5.1.1.76 fc2H264Open 72
5.1.1.77 fc2IsCameraControlable 72
5.1.1.78 fc2LaunchBrowser 72

5.1.1.79 fc2LaunchCommand 73
5.1.1.80 fc2LaunchCommandAsync 73
5.1.1.81 fc2LaunchHelp 73
5.1.1.82 fc2MJPGOpen 74
5.1.1.83 fc2QueryGigEImagingMode 74
5.1.1.84 fc2ReadGVCPMemory 74
5.1.1.85 fc2ReadGVCPRegister 74
5.1.1.86 fc2ReadGVCPRegisterBlock 75
5.1.1.87 fc2ReadRegister 75
5.1.1.88 fc2ReadRegisterBlock 75
5.1.1.89 fc2RegisterCallback 76
5.1.1.90 fc2RescanBus 76
5.1.1.91 fc2RestoreFromMemoryChannel 76
5.1.1.92 fc2RetrieveBuffer 77
5.1.1.93 fc2SaveImage 77
5.1.1.94 fc2SaveImageWithOption 77
5.1.1.95 fc2SaveToMemoryChannel 78
5.1.1.96 fc2SetActiveLUTBank 78
5.1.1.97 fc2SetCallback 78
5.1.1.98 fc2SetChannelStatus 79
5.1.1.99 fc2SetConfiguration 79
5.1.1.100 fc2SetDefaultColorProcessing 79
5.1.1.101 fc2SetDefaultOutputFormat 80
5.1.1.102 fc2SetEmbeddedImageInfo 80
5.1.1.103 fc2SetFormat7Configuration 81
5.1.1.104 fc2SetFormat7ConfigurationPacket 81
5.1.1.105 fc2SetGigEConfig 81
5.1.1.106 fc2SetGigEImageBinningSettings 81
5.1.1.107 fc2SetGigEImageSettings 81
5.1.1.108 fc2SetGigEImagingMode 82
5.1.1.109 fc2SetGigEProperty 82
5.1.1.110 fc2SetGigEStreamChannelInfo 82
5.1.1.111 fc2SetGPIOPinDirection 82
5.1.1.112 fc2SetGPIOPinDirectionBroadcast 82

5.1.1.113 fc2SetImageData 83
5.1.1.114 fc2SetImageDimensions 83
5.1.1.115 fc2SetLUTChannel 84
5.1.1.116 fc2SetProperty 84
5.1.1.117 fc2SetPropertyBroadcast 84
5.1.1.118 fc2SetStrobe 85
5.1.1.119 fc2SetStrobeBroadcast 85
5.1.1.120 fc2SetTriggerDelay 85
5.1.1.121 fc2SetTriggerDelayBroadcast 86
5.1.1.122 fc2SetTriggerMode 86
5.1.1.123 fc2SetTriggerModeBroadcast 86
5.1.1.124 fc2SetUserBuffers 87
5.1.1.125 fc2SetVideoModeAndFrameRate 87
5.1.1.126 fc2StartCapture 87
5.1.1.127 fc2StartCaptureCallback 88
5.1.1.128 fc2StartSyncCapture 88
5.1.1.129 fc2StartSyncCaptureCallback 88
5.1.1.130 fc2StopCapture 89
5.1.1.131 fc2UnregisterCallback 89
5.1.1.132 fc2ValidateFormat7Settings 89
5.1.1.133 fc2WriteGVCPMemory 90
5.1.1.134 fc2WriteGVCPRegister 90
5.1.1.135 fc2WriteGVCPRegisterBlock 91
5.1.1.136 fc2WriteGVCPRegisterBroadcast 91
5.1.1.137 fc2WriteRegister 91
5.1.1.138 fc2WriteRegisterBlock 92
5.1.1.139 fc2WriteRegisterBroadcast 92
5.2 FlyCapture2Defs_C.h File Reference 92
5.2.1 Define Documentation 97
5.2.1.1 FALSE 97
5.2.1.2 FULL_32BIT_VALUE 97
5.2.1.3 MAX_STRING_LENGTH 97
5.2.1.4 TRUE 97
5.2.2 Typedef Documentation 97

5.2.2.1 BOOL 97
5.2.2.2 fc2AsyncCommandCallback 97
5.2.2.3 fc2AVIContext 97
5.2.2.4 fc2BusEventCallback 97
5.2.2.5 fc2CallbackHandle 97
5.2.2.6 fc2Context 97
5.2.2.7 fc2GuiContext 97
5.2.2.8 fc2ImageEventCallback 98
5.2.2.9 fc2ImageImpl 98
5.2.2.10 fc2ImageStatisticsContext 98
5.2.3 Enumeration Type Documentation 98
5.2.3.1 fc2BandwidthAllocation 98
5.2.3.2 fc2BayerTileFormat 98
5.2.3.3 fc2BusCallbackType 98
5.2.3.4 fc2BusSpeed 99
5.2.3.5 fc2ByteOrder 99
5.2.3.6 fc2ColorProcessingAlgorithm 99
5.2.3.7 fc2DriverType 100
5.2.3.8 fc2Error 100
5.2.3.9 fc2FrameRate 102
5.2.3.10 fc2GigEPropertyType 102
5.2.3.11 fc2GrabMode 102
5.2.3.12 fc2GrabTimeout 102
5.2.3.13 fc2ImageFileFormat 103
5.2.3.14 fc2InterfaceType 103
5.2.3.15 fc2Mode 103
5.2.3.16 fc2OSType 104
5.2.3.17 fc2PCIeBusSpeed 105
5.2.3.18 fc2PixelFormat 105
5.2.3.19 fc2PropertyType 106
5.2.3.20 fc2StatisticsChannel 106
5.2.3.21 fc2TIFFCompressionMethod 107
5.2.3.22 fc2VideoMode 107
5.3 FlyCapture2GUI_C.h File Reference 108

5.3.1 Function Documentation 108
5.3.1.1 fc2CreateGUIContext 108
5.3.1.2 fc2DestroyGUIContext 108
5.3.1.3 fc2Disonnect 109
5.3.1.4 fc2GUIConnect 109
5.3.1.5 fc2GUIDisconnect 109
5.3.1.6 fc2Hide 110
5.3.1.7 fc2IsVisible 110
5.3.1.8 fc2Show 110
5.3.1.9 fc2ShowModal 110
5.4 FlyCapture2Internal_C.h File Reference 111
5.4.1 Function Documentation 111
5.4.1.1 IsContextValid 111
5.4.1.2 IsGuiContextValid 111
5.4.1.3 SyncCppImageToStruct 111
5.5 FlyCapture2Platform_C.h File Reference 111
5.5.1 Define Documentation 111
5.5.1.1 FLYCAPTURE2_C_API 111
5.5.1.2 FLYCAPTURE2_C_CALL_CONVEN 111
5.6 MultiSyncLibrary_C.h File Reference 111
5.6.1 Function Documentation 112
5.6.1.1 syncCreateContext 112
5.6.1.2 syncDestroyContext 113
5.6.1.3 syncDisableCrossPCSynchronization 113
5.6.1.4 syncEnableCrossPCSynchronization 113
5.6.1.5 syncGetStatus 114
5.6.1.6 syncGetTimeSinceSynced 114
5.6.1.7 syncIsTimingBusConnected 114
5.6.1.8 syncQueryCrossPCSynchronizationSetting 114
5.6.1.9 syncRescanMasterTimingBus 115
5.6.1.10 syncStart 115
5.6.1.11 syncStop 115
5.7 MultiSyncLibraryDefs_C.h File Reference 115
5.7.1 Define Documentation 116

5.7.1.1 FALSE 116
5.7.1.2 FULL_32BIT_VALUE 116
5.7.1.3 MAX_STRING_LENGTH 116
5.7.1.4 TRUE 116
5.7.2 Typedef Documentation 116
5.7.2.1 BOOL 116
5.7.2.2 syncContext 116
5.7.3 Enumeration Type Documentation 117
5.7.3.1 syncError 117
5.7.3.2 syncMessage 117
5.8 MultiSyncLibraryPlatform_C.h File Reference 117
5.8.1 Define Documentation 117
5.8.1.1 MULTISYNCLIBRARY_C_API 117
5.8.1.2 MULTISYNCLIBRARY_C_CALL_CONVEN 117

Chapter 1

Deprecated List

Global fc2Disonnect (fc2GuiContext context) __attribute__((deprecated))

This method is deprecated and will be removed in a future FlyCapture2 release. Please use fc2GUIDisconnect instead.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

fc2AVIOption 7
fc2CameraInfo 8
fc2Config 10
fc2ConfigROM 11
fc2EmbeddedImageInfo 13
fc2EmbeddedImageInfoProperty 14
fc2Format7ImageSettings 14
fc2Format7Info 15
fc2Format7PacketInfo 16
fc2GigEConfig 17
fc2GigEImageSettings 18
fc2GigEImageSettingsInfo 18
fc2GigEProperty 19
fc2GigEStreamChannel 20
fc2H264Option 21
fc2Image 22
fc2ImageMetadata 22
fc2InternalContext 23
fc2InternalGuiContext 24
fc2InternalImageCallback 25
fc2IPAddress 25
fc2JPEGOption 26
fc2JPG2Option 26
fc2LUTData 27
fc2MACAddress 27
fc2MJPGOption 28
fc2PGMOption 28
fc2PGRGuid
A GUID to the camera 29

fc2PNGOption 29
fc2PPMOption 30
fc2StrobeControl 30
fc2StrobeInfo 31
fc2SystemInfo 31
fc2TIFFOption 32
fc2TimeStamp 33
fc2TriggerDelay 33
fc2TriggerDelayInfo 34
fc2TriggerMode 36
fc2TriggerModeInfo 36
fc2Version 37

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

FlyCapture2_C.h 39
FlyCapture2Defs_C.h 92
FlyCapture2GUI_C.h 108
FlyCapture2Internal_C.h 111
FlyCapture2Platform_C.h 111
MultiSyncLibrary_C.h 111
MultiSyncLibraryDefs_C.h 115
MultiSyncLibraryPlatform_C.h 117

Chapter 4

Data Structure Documentation

4.1 fc2AVIOption Struct Reference

Data Fields

- float frameRate
- unsigned int reserved [256]

4.1.1 Field Documentation

4.1.1.1 float frameRate

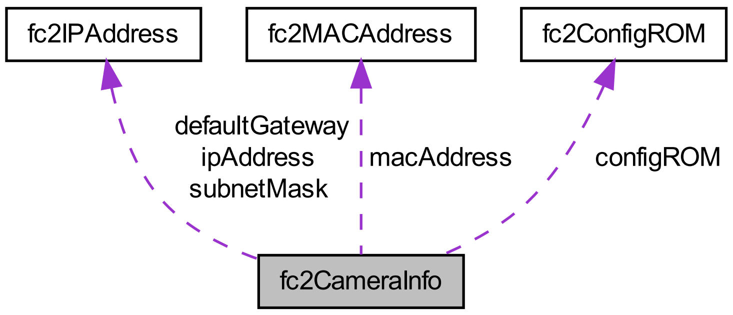
4.1.1.2 unsigned int reserved[256]

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

- FlyCapture2Defs_C.h

4.2 fc2CameraInfo Struct Reference

Collaboration diagram for fc2CameraInfo:



Data Fields

- unsigned int serialNumber
- fc2InterfaceType interfaceType
- fc2DriverType driverType
- BOOL isColorCamera
- char modelName [MAX_STRING_LENGTH]
- char vendorName [MAX_STRING_LENGTH]
- char sensorInfo [MAX_STRING_LENGTH]
- char sensorResolution [MAX_STRING_LENGTH]
- char driverName [MAX_STRING_LENGTH]
- char firmwareVersion [MAX_STRING_LENGTH]
- char firmwareBuildTime [MAX_STRING_LENGTH]
- fc2BusSpeed maximumBusSpeed
- fc2PCIeBusSpeed pcieBusSpeed
- fc2BayerTileFormat bayerTileFormat
- unsigned short busNumber
- unsigned short nodeNumber
- unsigned int iidcVer
- fc2ConfigROM configROM
- unsigned int gigEMajorVersion
- unsigned int gigEMinorVersion
- char userDefinedName [MAX_STRING_LENGTH]
- char xmlURL1 [MAX_STRING_LENGTH]
- char xmlURL2 [MAX_STRING_LENGTH]

- fc2MACAddress macAddress
- fc2IPAddress ipAddress
- fc2IPAddress subnetMask
- fc2IPAddress defaultGateway
- unsigned int ccpStatus

  Status/Content of CCP register.
- unsigned int applicationIPAddress

  Local Application IP Address.
- unsigned int applicationPort

  Local Application port.
- unsigned int reserved [16]

4.2.1 Field Documentation

4.2.1.1 unsigned int **applicationIPAddress**

Local Application IP Address.

4.2.1.2 unsigned int **applicationPort**

Local Application port.

4.2.1.3 fc2BayerTileFormat bayerTileFormat

4.2.1.4 unsigned short **busNumber**

4.2.1.5 unsigned int **ccpStatus**

Status/Content of CCP register.

4.2.1.6 fc2ConfigROM configROM

4.2.1.7 fc2IPAddress defaultGateway

4.2.1.8 char **driverName**[MAX_STRING_LENGTH]

4.2.1.9 fc2DriverType driverType

4.2.1.10 char **firmwareBuildTime**[MAX_STRING_LENGTH]

4.2.1.11 char **firmwareVersion**[MAX_STRING_LENGTH]

4.2.1.12 unsigned int **gigEMajorVersion**

4.2.1.13 unsigned int **gigEMinorVersion**

4.2.1.14 unsigned int **iidcVer**

4.2.1.15 **fc2InterfaceType interfaceType**

4.2.1.16 **fc2IPAddress ipAddress**

4.2.1.17 **BOOL isColorCamera**

4.2.1.18 **fc2MACAddress macAddress**

4.2.1.19 **fc2BusSpeed maximumBusSpeed**

4.2.1.20 char **modelName**[MAX_STRING_LENGTH]

4.2.1.21 unsigned short **nodeNumber**

4.2.1.22 **fc2PCIeBusSpeed pcieBusSpeed**

4.2.1.23 unsigned int **reserved**[16]

4.2.1.24 char **sensorInfo**[MAX_STRING_LENGTH]

4.2.1.25 char **sensorResolution**[MAX_STRING_LENGTH]

4.2.1.26 unsigned int **serialNumber**

4.2.1.27 **fc2IPAddress subnetMask**

4.2.1.28 char **userDefinedName**[MAX_STRING_LENGTH]

4.2.1.29 char **vendorName**[MAX_STRING_LENGTH]

4.2.1.30 char **xmlURL1**[MAX_STRING_LENGTH]

4.2.1.31 char **xmlURL2**[MAX_STRING_LENGTH]

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

- FlyCapture2Defs_C.h

4.3 fc2Config Struct Reference

Data Fields

- unsigned int numBuffers

- unsigned int numImageNotifications
- unsigned int minNumImageNotifications
- int grabTimeout
- fc2GrabMode grabMode
- fc2BusSpeed isochBusSpeed
- fc2BusSpeed asyncBusSpeed
- fc2BandwidthAllocation bandwidthAllocation
- unsigned int registerTimeoutRetries
- unsigned int registerTimeout
- unsigned int reserved [16]

4.3.1 Field Documentation

4.3.1.1 fc2BusSpeed asyncBusSpeed

4.3.1.2 fc2BandwidthAllocation bandwidthAllocation

4.3.1.3 fc2GrabMode grabMode

4.3.1.4 int grabTimeout

4.3.1.5 fc2BusSpeed isochBusSpeed

4.3.1.6 unsigned int minNumImageNotifications

4.3.1.7 unsigned int numBuffers

4.3.1.8 unsigned int numImageNotifications

4.3.1.9 unsigned int registerTimeout

4.3.1.10 unsigned int registerTimeoutRetries

4.3.1.11 unsigned int reserved[16]

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

- FlyCapture2Defs_C.h

4.4 fc2ConfigROM Struct Reference

Data Fields

- unsigned int nodeVendorId
- unsigned int chipIdHi

- unsigned int chipIdLo
- unsigned int unitSpecId
- unsigned int unitSWVer
- unsigned int unitSubSWVer
- unsigned int vendorUniqueInfo_0
- unsigned int vendorUniqueInfo_1
- unsigned int vendorUniqueInfo_2
- unsigned int vendorUniqueInfo_3
- char pszKeyword [MAX_STRING_LENGTH]
- unsigned int reserved [16]

4.4.1 Field Documentation

4.4.1.1 unsigned int **chipIdHi**

4.4.1.2 unsigned int **chipIdLo**

4.4.1.3 unsigned int **nodeVendorId**

4.4.1.4 char **pszKeyword**[MAX_STRING_LENGTH]

4.4.1.5 unsigned int **reserved**[16]

4.4.1.6 unsigned int **unitSpecId**

4.4.1.7 unsigned int **unitSubSWVer**

4.4.1.8 unsigned int **unitSWVer**

4.4.1.9 unsigned int **vendorUniqueInfo_0**

4.4.1.10 unsigned int **vendorUniqueInfo_1**

4.4.1.11 unsigned int **vendorUniqueInfo_2**

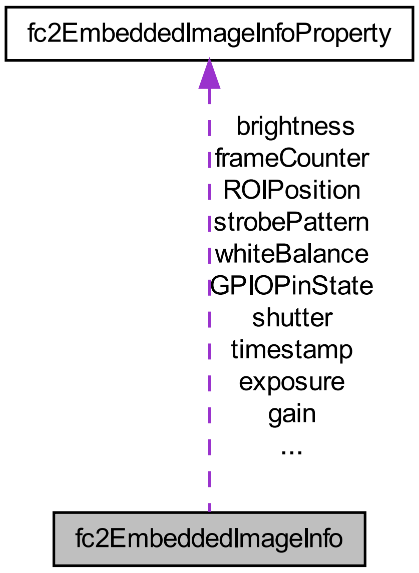
4.4.1.12 unsigned int **vendorUniqueInfo_3**

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

- FlyCapture2Defs_C.h

4.5 fc2EmbeddedImageInfo Struct Reference

Collaboration diagram for fc2EmbeddedImageInfo:



Data Fields

- fc2EmbeddedImageInfoProperty timestamp
- fc2EmbeddedImageInfoProperty gain
- fc2EmbeddedImageInfoProperty shutter
- fc2EmbeddedImageInfoProperty brightness
- fc2EmbeddedImageInfoProperty exposure
- fc2EmbeddedImageInfoProperty whiteBalance
- fc2EmbeddedImageInfoProperty frameCounter
- fc2EmbeddedImageInfoProperty strobePattern
- fc2EmbeddedImageInfoProperty GPIOPinState
- fc2EmbeddedImageInfoProperty ROIPosition

4.5.1 Field Documentation

4.5.1.1 fc2EmbeddedImageInfoProperty brightness

4.5.1.2 fc2EmbeddedImageInfoProperty exposure

4.5.1.3 fc2EmbeddedImageInfoProperty frameCounter

4.5.1.4 fc2EmbeddedImageInfoProperty gain

4.5.1.5 fc2EmbeddedImageInfoProperty GPIOPinState

4.5.1.6 fc2EmbeddedImageInfoProperty ROIPosition

4.5.1.7 fc2EmbeddedImageInfoProperty shutter

4.5.1.8 fc2EmbeddedImageInfoProperty strobePattern

4.5.1.9 fc2EmbeddedImageInfoProperty timestamp

4.5.1.10 fc2EmbeddedImageInfoProperty whiteBalance

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

- FlyCapture2Defs_C.h

4.6 fc2EmbeddedImageInfoProperty Struct Reference

Data Fields

- BOOL available
- BOOL onOff

4.6.1 Field Documentation

4.6.1.1 BOOL available

4.6.1.2 BOOL onOff

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

- FlyCapture2Defs_C.h

4.7 fc2Format7ImageSettings Struct Reference

Data Fields

- fc2Mode mode

- unsigned int offsetX
- unsigned int offsetY
- unsigned int width
- unsigned int height
- fc2PixelFormat pixelFormat
- unsigned int reserved [8]

4.7.1 Field Documentation

4.7.1.1 unsigned int height

4.7.1.2 fc2Mode mode

4.7.1.3 unsigned int offsetX

4.7.1.4 unsigned int offsetY

4.7.1.5 fc2PixelFormat pixelFormat

4.7.1.6 unsigned int reserved[8]

4.7.1.7 unsigned int width

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

- FlyCapture2Defs_C.h

4.8 fc2Format7Info Struct Reference

Data Fields

- fc2Mode mode
- unsigned int maxWidth
- unsigned int maxHeight
- unsigned int offsetHStepSize
- unsigned int offsetVStepSize
- unsigned int imageHStepSize
- unsigned int imageVStepSize
- unsigned int pixelFormatBitField
- unsigned int vendorPixelFormatBitField
- unsigned int packetSize
- unsigned int minPacketSize
- unsigned int maxPacketSize
- float percentage
- unsigned int reserved [16]

4.8.1 Field Documentation

4.8.1.1 unsigned int **imageHStepSize**

4.8.1.2 unsigned int **imageVStepSize**

4.8.1.3 unsigned int **maxHeight**

4.8.1.4 unsigned int **maxPacketSize**

4.8.1.5 unsigned int **maxWidth**

4.8.1.6 unsigned int **minPacketSize**

4.8.1.7 **fc2Mode mode**

4.8.1.8 unsigned int **offsetHStepSize**

4.8.1.9 unsigned int **offsetVStepSize**

4.8.1.10 unsigned int **packetSize**

4.8.1.11 float **percentage**

4.8.1.12 unsigned int **pixelFormatBitField**

4.8.1.13 unsigned int **reserved**[16]

4.8.1.14 unsigned int **vendorPixelFormatBitField**

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

- FlyCapture2Defs_C.h

4.9 fc2Format7PacketInfo Struct Reference

Data Fields

- unsigned int recommendedBytesPerPacket
- unsigned int maxBytesPerPacket
- unsigned int unitBytesPerPacket
- unsigned int reserved [8]

4.9.1 Field Documentation

4.9.1.1 unsigned int **maxBytesPerPacket**

4.9.1.2 unsigned int **recommendedBytesPerPacket**

4.9.1.3 unsigned int **reserved**[8]

4.9.1.4 unsigned int **unitBytesPerPacket**

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

- FlyCapture2Defs_C.h

4.10 fc2GigEConfig Struct Reference

Data Fields

- BOOL enablePacketResend

  Turn on/off packet resend functionality.
- unsigned int timeoutForPacketResend

  The number of miliseconds to wait for each requested packet.
- unsigned int maxPacketsToResend

  The max number of packets that can be requested to be resend.
- unsigned int reserved [8]

4.10.1 Field Documentation

4.10.1.1 BOOL **enablePacketResend**

Turn on/off packet resend functionality.

4.10.1.2 unsigned int **maxPacketsToResend**

The max number of packets that can be requested to be resend.

4.10.1.3 unsigned int **reserved**[8]

4.10.1.4 unsigned int **timeoutForPacketResend**

The number of miliseconds to wait for each requested packet.

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

- FlyCapture2Defs_C.h

4.11 fc2GigEImageSettings Struct Reference

Data Fields

- unsigned int offsetX
- unsigned int offsetY
- unsigned int width
- unsigned int height
- fc2PixelFormat pixelFormat
- unsigned int reserved [8]

4.11.1 Field Documentation

4.11.1.1 unsigned int **height**

4.11.1.2 unsigned int **offsetX**

4.11.1.3 unsigned int **offsetY**

4.11.1.4 fc2PixelFormat **pixelFormat**

4.11.1.5 unsigned int **reserved**[8]

4.11.1.6 unsigned int **width**

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

- FlyCapture2Defs_C.h

4.12 fc2GigEImageSettingsInfo Struct Reference

Data Fields

- unsigned int maxWidth
- unsigned int maxHeight
- unsigned int offsetHStepSize
- unsigned int offsetVStepSize
- unsigned int imageHStepSize
- unsigned int imageVStepSize
- unsigned int pixelFormatBitField
- unsigned int vendorPixelFormatBitField
- unsigned int reserved [16]

4.12.1 Field Documentation

4.12.1.1 unsigned int imageHStepSize

4.12.1.2 unsigned int imageVStepSize

4.12.1.3 unsigned int maxHeight

4.12.1.4 unsigned int maxWidth

4.12.1.5 unsigned int offsetHStepSize

4.12.1.6 unsigned int offsetVStepSize

4.12.1.7 unsigned int pixelFormatBitField

4.12.1.8 unsigned int reserved[16]

4.12.1.9 unsigned int vendorPixelFormatBitField

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

- FlyCapture2Defs_C.h

4.13 fc2GigEProperty Struct Reference

Data Fields

- fc2GigEPropertyType propType
- BOOL isReadable
- BOOL isWritable
- unsigned int min
- unsigned int max
- unsigned int value
- unsigned int reserved [8]

4.13.1 Field Documentation

4.13.1.1 BOOL isReadable

4.13.1.2 BOOL isWritable

4.13.1.3 unsigned int max

4.13.1.4 unsigned int min

4.13.1.5 fc2GigEPropertyType propType

4.13.1.6 unsigned int reserved[8]

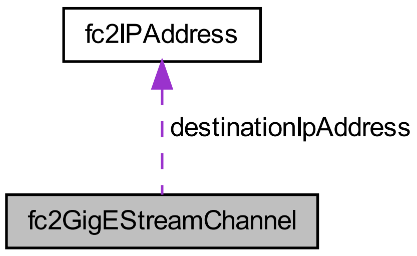
4.13.1.7 unsigned int value

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

- FlyCapture2Defs_C.h

4.14 fc2GigEStreamChannel Struct Reference

Collaboration diagram for fc2GigEStreamChannel:



Data Fields

- unsigned int networkInterfaceIndex
- unsigned int hostPost
- BOOL doNotFragment
- unsigned int packetSize
- unsigned int interPacketDelay
- fc2IPAddress destinationIpAddress
- unsigned int sourcePort
- unsigned int reserved [8]

4.14.1 Field Documentation

4.14.1.1 fc2IPAddress destinationIpAddress

4.14.1.2 BOOL doNotFragment

4.14.1.3 unsigned int hostPost

4.14.1.4 unsigned int interPacketDelay

4.14.1.5 unsigned int networkInterfaceIndex

4.14.1.6 unsigned int packetSize

4.14.1.7 unsigned int reserved[8]

4.14.1.8 unsigned int sourcePort

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

- FlyCapture2Defs_C.h

4.15 fc2H264Option Struct Reference

Data Fields

- float frameRate
- unsigned int width
- unsigned int height
- unsigned int bitrate
- unsigned int reserved [256]

4.15.1 Field Documentation

4.15.1.1 unsigned int bitrate

4.15.1.2 float frameRate

4.15.1.3 unsigned int height

4.15.1.4 unsigned int reserved[256]

4.15.1.5 unsigned int width

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

- FlyCapture2Defs_C.h

4.16 fc2Image Struct Reference

Data Fields

- unsigned int rows
- unsigned int cols
- unsigned int stride
- unsigned char * pData
- unsigned int dataSize
- unsigned int receivedDataSize
- fc2PixelFormat format
- fc2BayerTileFormat bayerFormat
- fc2ImageImpl imageImpl

4.16.1 Field Documentation

4.16.1.1 fc2BayerTileFormat bayerFormat

4.16.1.2 unsigned int cols

4.16.1.3 unsigned int dataSize

4.16.1.4 fc2PixelFormat format

4.16.1.5 fc2ImageImpl imageImpl

4.16.1.6 unsigned char* pData

4.16.1.7 unsigned int receivedDataSize

4.16.1.8 unsigned int rows

4.16.1.9 unsigned int stride

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

- FlyCapture2Defs_C.h

4.17 fc2ImageMetadata Struct Reference

Data Fields

- unsigned int embeddedTimeStamp
- unsigned int embeddedGain
- unsigned int embeddedShutter

- unsigned int embeddedBrightness
- unsigned int embeddedExposure
- unsigned int embeddedWhiteBalance
- unsigned int embeddedFrameCounter
- unsigned int embeddedStrobePattern
- unsigned int embeddedGPIOPinState
- unsigned int embeddedROIPosition
- unsigned int reserved [31]

4.17.1 Field Documentation

4.17.1.1 unsigned int **embeddedBrightness**

4.17.1.2 unsigned int **embeddedExposure**

4.17.1.3 unsigned int **embeddedFrameCounter**

4.17.1.4 unsigned int **embeddedGain**

4.17.1.5 unsigned int **embeddedGPIOPinState**

4.17.1.6 unsigned int **embeddedROIPosition**

4.17.1.7 unsigned int **embeddedShutter**

4.17.1.8 unsigned int **embeddedStrobePattern**

4.17.1.9 unsigned int **embeddedTimeStamp**

4.17.1.10 unsigned int **embeddedWhiteBalance**

4.17.1.11 unsigned int **reserved**[31]

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

- FlyCapture2Defs_C.h

4.18 fc2InternalContext Struct Reference

Data Fields

- FlyCapture2::BusManager ∗ pBusMgr
- FlyCapture2::CameraBase ∗ pCamera

4.18.1 Field Documentation

4.18.1.1 FlyCapture2::BusManager∗ pBusMgr

4.18.1.2 FlyCapture2::CameraBase∗ pCamera

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

- FlyCapture2Internal_C.h

4.19 fc2InternalGuiContext Struct Reference

Data Fields

- FlyCapture2::CameraSelectionDlg ∗ pCameraSelectionDlg
- FlyCapture2::CameraControlDlg ∗ pCameraControlDlg

4.19.1 Field Documentation

4.19.1.1 FlyCapture2::CameraControlDlg∗ pCameraControlDlg

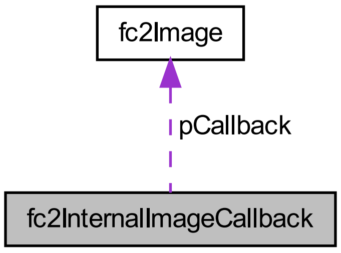
4.19.1.2 FlyCapture2::CameraSelectionDlg∗ pCameraSelectionDlg

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

- FlyCapture2Internal_C.h

4.20 fc2InternalImageCallback Struct Reference

Collaboration diagram for fc2InternalImageCallback:



Data Fields

- fc2ImageEventCallback pCallback
- void ∗ pCallbackData

4.20.1 Field Documentation

4.20.1.1 fc2ImageEventCallback pCallback

4.20.1.2 void∗ pCallbackData

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

- FlyCapture2Internal_C.h

4.21 fc2IPAddress Struct Reference

Data Fields

- unsigned char octets [4]

4.21.1 Field Documentation

4.21.1.1 unsigned char **octets**[4]

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

- FlyCapture2Defs_C.h

4.22 fc2JPEGOption Struct Reference

Data Fields

- BOOL progressive
- unsigned int quality
- unsigned int reserved [16]

4.22.1 Field Documentation

4.22.1.1 **BOOL progressive**

4.22.1.2 unsigned int **quality**

4.22.1.3 unsigned int **reserved**[16]

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

- FlyCapture2Defs_C.h

4.23 fc2JPG2Option Struct Reference

Data Fields

- unsigned int quality
- unsigned int reserved [16]

4.23.1 Field Documentation

4.23.1.1 unsigned int **quality**

4.23.1.2 unsigned int **reserved**[16]

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

- FlyCapture2Defs_C.h

4.24 fc2LUTData Struct Reference

Data Fields

- BOOL supported
- BOOL enabled
- unsigned int numBanks
- unsigned int numChannels
- unsigned int inputBitDepth
- unsigned int outputBitDepth
- unsigned int numEntries
- unsigned int reserved [8]

4.24.1 Field Documentation

4.24.1.1 BOOL enabled

4.24.1.2 unsigned int inputBitDepth

4.24.1.3 unsigned int numBanks

4.24.1.4 unsigned int numChannels

4.24.1.5 unsigned int numEntries

4.24.1.6 unsigned int outputBitDepth

4.24.1.7 unsigned int reserved[8]

4.24.1.8 BOOL supported

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

- FlyCapture2Defs_C.h

4.25 fc2MACAddress Struct Reference

Data Fields

- unsigned char octets [6]

4.25.1 Field Documentation

4.25.1.1 unsigned char **octets**[6]

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

- FlyCapture2Defs_C.h

4.26 fc2MJPGOption Struct Reference

Data Fields

- float frameRate
- unsigned int quality
- unsigned int reserved [256]

4.26.1 Field Documentation

4.26.1.1 float **frameRate**

4.26.1.2 unsigned int **quality**

4.26.1.3 unsigned int **reserved**[256]

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

- FlyCapture2Defs_C.h

4.27 fc2PGMOption Struct Reference

Data Fields

- BOOL binaryFile
- unsigned int reserved [16]

4.27.1 Field Documentation

4.27.1.1 **BOOL binaryFile**

4.27.1.2 unsigned int **reserved**[16]

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

- FlyCapture2Defs_C.h

4.28 fc2PGRGuid Struct Reference

A GUID to the camera.

Data Fields

- unsigned int value [4]

4.28.1 Detailed Description

A GUID to the camera.

It is used to uniquely identify a camera.

4.28.2 Field Documentation

4.28.2.1 unsigned int **value**[4]

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

- FlyCapture2Defs_C.h

4.29 fc2PNGOption Struct Reference

Data Fields

- BOOL interlaced
- unsigned int compressionLevel
- unsigned int reserved [16]

4.29.1 Field Documentation

4.29.1.1 unsigned int **compressionLevel**

4.29.1.2 **BOOL interlaced**

4.29.1.3 unsigned int **reserved**[16]

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

- FlyCapture2Defs_C.h

4.30 fc2PPMOption Struct Reference

Data Fields

- BOOL binaryFile
- unsigned int reserved [16]

4.30.1 Field Documentation

4.30.1.1 BOOL binaryFile

4.30.1.2 unsigned int reserved[16]

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

- FlyCapture2Defs_C.h

4.31 fc2StrobeControl Struct Reference

Data Fields

- unsigned int source
- BOOL onOff
- unsigned int polarity
- float delay
- float duration
- unsigned int reserved [8]

4.31.1 Field Documentation

4.31.1.1 float delay

4.31.1.2 float duration

4.31.1.3 BOOL onOff

4.31.1.4 unsigned int polarity

4.31.1.5 unsigned int reserved[8]

4.31.1.6 unsigned int source

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

- FlyCapture2Defs_C.h

4.32 fc2StrobeInfo Struct Reference

Data Fields

- unsigned int source
- BOOL present
- BOOL readOutSupported
- BOOL onOffSupported
- BOOL polaritySupported
- float minValue
- float maxValue
- unsigned int reserved [8]

4.32.1 Field Documentation

4.32.1.1 float maxValue

4.32.1.2 float minValue

4.32.1.3 BOOL onOffSupported

4.32.1.4 BOOL polaritySupported

4.32.1.5 BOOL present

4.32.1.6 BOOL readOutSupported

4.32.1.7 unsigned int reserved[8]

4.32.1.8 unsigned int source

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

- FlyCapture2Defs_C.h

4.33 fc2SystemInfo Struct Reference

Data Fields

- fc2OSType osType
- char osDescription [MAX_STRING_LENGTH]
- fc2ByteOrder byteOrder
- size_t sysMemSize
- char cpuDescription [MAX_STRING_LENGTH]
- size_t numCpuCores

- char driverList [MAX_STRING_LENGTH]
- char libraryList [MAX_STRING_LENGTH]
- char gpuDescription [MAX_STRING_LENGTH]
- size_t screenWidth
- size_t screenHeight
- unsigned int reserved [16]

4.33.1 Field Documentation

4.33.1.1 fc2ByteOrder byteOrder

4.33.1.2 char cpuDescription[MAX_STRING_LENGTH]

4.33.1.3 char driverList[MAX_STRING_LENGTH]

4.33.1.4 char gpuDescription[MAX_STRING_LENGTH]

4.33.1.5 char libraryList[MAX_STRING_LENGTH]

4.33.1.6 size_t numCpuCores

4.33.1.7 char osDescription[MAX_STRING_LENGTH]

4.33.1.8 fc2OSType osType

4.33.1.9 unsigned int reserved[16]

4.33.1.10 size_t screenHeight

4.33.1.11 size_t screenWidth

4.33.1.12 size_t sysMemSize

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

- FlyCapture2Defs_C.h

4.34 fc2TIFFOption Struct Reference

Data Fields

- fc2TIFFCompressionMethod compression
- unsigned int reserved [16]

4.34.1 Field Documentation

4.34.1.1 fc2TIFFCompressionMethod compression

4.34.1.2 unsigned int reserved[16]

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

- FlyCapture2Defs_C.h

4.35 fc2TimeStamp Struct Reference

Data Fields

- long long seconds
- unsigned int microSeconds
- unsigned int cycleSeconds
- unsigned int cycleCount
- unsigned int cycleOffset
- unsigned int reserved [8]

4.35.1 Field Documentation

4.35.1.1 unsigned int cycleCount

4.35.1.2 unsigned int cycleOffset

4.35.1.3 unsigned int cycleSeconds

4.35.1.4 unsigned int microSeconds

4.35.1.5 unsigned int reserved[8]

4.35.1.6 long long seconds

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

- FlyCapture2Defs_C.h

4.36 fc2TriggerDelay Struct Reference

Data Fields

- fc2PropertyType type

- BOOL present
- BOOL absControl
- BOOL onePush
- BOOL onOff
- BOOL autoManualMode
- unsigned int valueA
- unsigned int valueB
- float absValue
- unsigned int reserved [8]

4.36.1 Field Documentation

4.36.1.1 BOOL absControl

4.36.1.2 float absValue

4.36.1.3 BOOL autoManualMode

4.36.1.4 BOOL onePush

4.36.1.5 BOOL onOff

4.36.1.6 BOOL present

4.36.1.7 unsigned int reserved[8]

4.36.1.8 fc2PropertyType type

4.36.1.9 unsigned int valueA

4.36.1.10 unsigned int valueB

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

- FlyCapture2Defs_C.h

4.37 fc2TriggerDelayInfo Struct Reference

Data Fields

- fc2PropertyType type
- BOOL present
- BOOL autoSupported
- BOOL manualSupported
- BOOL onOffSupported

- BOOL onePushSupported
- BOOL absValSupported
- BOOL readOutSupported
- unsigned int min
- unsigned int max
- float absMin
- float absMax
- char pUnits [MAX_STRING_LENGTH]
- char pUnitAbbr [MAX_STRING_LENGTH]
- unsigned int reserved [8]

4.37.1 Field Documentation

4.37.1.1 float absMax

4.37.1.2 float absMin

4.37.1.3 BOOL absValSupported

4.37.1.4 BOOL autoSupported

4.37.1.5 BOOL manualSupported

4.37.1.6 unsigned int max

4.37.1.7 unsigned int min

4.37.1.8 BOOL onePushSupported

4.37.1.9 BOOL onOffSupported

4.37.1.10 BOOL present

4.37.1.11 char pUnitAbbr[MAX_STRING_LENGTH]

4.37.1.12 char pUnits[MAX_STRING_LENGTH]

4.37.1.13 BOOL readOutSupported

4.37.1.14 unsigned int reserved[8]

4.37.1.15 fc2PropertyType type

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

- FlyCapture2Defs_C.h

4.38 fc2TriggerMode Struct Reference

Data Fields

- BOOL onOff
- unsigned int polarity
- unsigned int source
- unsigned int mode
- unsigned int parameter
- unsigned int reserved [8]

4.38.1 Field Documentation

4.38.1.1 unsigned int **mode**

4.38.1.2 BOOL onOff

4.38.1.3 unsigned int **parameter**

4.38.1.4 unsigned int **polarity**

4.38.1.5 unsigned int **reserved**[8]

4.38.1.6 unsigned int **source**

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

- FlyCapture2Defs_C.h

4.39 fc2TriggerModeInfo Struct Reference

Data Fields

- BOOL present
- BOOL readOutSupported
- BOOL onOffSupported
- BOOL polaritySupported
- BOOL valueReadable
- unsigned int sourceMask
- BOOL softwareTriggerSupported
- unsigned int modeMask
- unsigned int reserved [8]

4.39.1 Field Documentation

4.39.1.1 unsigned int **modeMask**

4.39.1.2 BOOL **onOffSupported**

4.39.1.3 BOOL **polaritySupported**

4.39.1.4 BOOL **present**

4.39.1.5 BOOL **readOutSupported**

4.39.1.6 unsigned int **reserved**[8]

4.39.1.7 BOOL **softwareTriggerSupported**

4.39.1.8 unsigned int **sourceMask**

4.39.1.9 BOOL **valueReadable**

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

- FlyCapture2Defs_C.h

4.40 fc2Version Struct Reference

Data Fields

- unsigned int major
- unsigned int minor
- unsigned int type
- unsigned int build

4.40.1 Field Documentation

4.40.1.1 unsigned int **build**

4.40.1.2 unsigned int **major**

4.40.1.3 unsigned int **minor**

4.40.1.4 unsigned int **type**

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

- FlyCapture2Defs_C.h

Chapter 5

File Documentation

5.1 FlyCapture2_C.h File Reference

Functions

- FLYCAPTURE2_C_API fc2Error fc2CreateContext (fc2Context ∗pContext)

    Create a FC2 context for IIDC camaera.
- FLYCAPTURE2_C_API fc2Error fc2CreateGigEContext (fc2Context ∗pContext)

    Create a FC2 context for a GigE Vision camera.
- FLYCAPTURE2_C_API fc2Error fc2DestroyContext (fc2Context context)

    Destroy the FC2 context.
- FLYCAPTURE2_C_API fc2Error fc2FireBusReset (fc2Context context, fc2PGRGuid ∗pGuid)

    Fire a bus reset.
- FLYCAPTURE2_C_API fc2Error fc2GetNumOfCameras (fc2Context context, unsigned int ∗pNumCameras)

    Gets the number of cameras attached to the PC.
- FLYCAPTURE2_C_API fc2Error fc2IsCameraControlable (fc2Context context, fc2PGRGuid ∗pGuid, BOOL ∗pControlable)

    Query whether a GigE camera is controlable.
- FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIndex (fc2Context context, unsigned int index, fc2PGRGuid ∗pGuid)

    Gets the PGRGuid for a camera on the PC.
- FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIPAddress (fc2Context context, fc2IPAddress ipAddress, fc2PGRGuid ∗pGuid)

    Gets the PGRGuid for a camera with the specified IPv4 address.
- FLYCAPTURE2_C_API fc2Error fc2GetCameraFromSerialNumber (fc2Context context, unsigned int serialNumber, fc2PGRGuid ∗pGuid)

    Gets the PGRGuid for a camera on the PC.

- FLYCAPTURE2_C_API fc2Error fc2GetCameraSerialNumberFromIndex (fc2Context context, unsigned int index, unsigned int ∗pSerialNumber)

  Gets the serial number of the camera with the specified index.
- FLYCAPTURE2_C_API fc2Error fc2GetInterfaceTypeFromGuid (fc2Context context, fc2PGRGuid ∗pGuid, fc2InterfaceType ∗pInterfaceType)

  Gets the interface type associated with a PGRGuid.
- FLYCAPTURE2_C_API fc2Error fc2GetNumOfDevices (fc2Context context, unsigned int ∗pNumDevices)

  Gets the number of devices.
- FLYCAPTURE2_C_API fc2Error fc2GetDeviceFromIndex (fc2Context context, unsigned int index, fc2PGRGuid ∗pGuid)

  Gets the PGRGuid for a device.
- FLYCAPTURE2_C_API fc2Error fc2RegisterCallback (fc2Context context, fc2BusEventCallback enumCallback, fc2BusCallbackType callbackType, void ∗pParameter, fc2CallbackHandle ∗pCallbackHandle)

  Register a callback function that will be called when the specified callback event occurs.
- FLYCAPTURE2_C_API fc2Error fc2UnregisterCallback (fc2Context context, fc2CallbackHandle callbackHandle)

  Unregister a callback function.
- FLYCAPTURE2_C_API fc2Error fc2RescanBus (fc2Context context)

  Force a rescan of the buses.
- FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressToCamera (fc2Context context, fc2MACAddress macAddress, fc2IPAddress ipAddress, fc2IPAddress subnetMask, fc2IPAddress defaultGateway)

  Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.
- FLYCAPTURE2_C_API fc2Error fc2ForceAllIPAddressesAutomatically ()

  Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the netowrk adapters that they are connected to.
- FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressAutomatically (unsigned int serialNumber)

  Force a cameras on the network to be assigned sequential IP addresses on the same subnet as the netowrk adapters that it is connected to.
- FLYCAPTURE2_C_API fc2Error fc2DiscoverGigECameras (fc2Context context, fc2CameraInfo ∗gigECameras, unsigned int ∗arraySize)

  Discover all cameras connected to the network even if they reside on a different subnet.
- FLYCAPTURE2_C_API fc2Error fc2WriteRegister (fc2Context context, unsigned int address, unsigned int value)

  Write to the specified register on the camera.
- FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBroadcast (fc2Context context, unsigned int address, unsigned int value)

  Write to the specified register on the camera with broadcast.
- FLYCAPTURE2_C_API fc2Error fc2ReadRegister (fc2Context context, unsigned int address, unsigned int ∗pValue)

  Read the specified register from the camera.

- FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBlock (fc2Context context, unsigned short addressHigh, unsigned int addressLow, const unsigned int ∗pBuffer, unsigned int length)

  Write to the specified register block on the camera.
- FLYCAPTURE2_C_API fc2Error fc2ReadRegisterBlock (fc2Context context, unsigned short addressHigh, unsigned int addressLow, unsigned int ∗pBuffer, unsigned int length)

  Write to the specified register block on the camera.
- FLYCAPTURE2_C_API fc2Error fc2Connect (fc2Context context, fc2PGRGuid ∗guid)

  Connects the camera object to the camera specified by the GUID.
- FLYCAPTURE2_C_API fc2Error fc2Disconnect (fc2Context context)

  Disconnects the fc2Context from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetCallback (fc2Context context, fc2ImageEventCallback pCallbackFn, void ∗pCallbackData)

  Sets the callback data to be used on completion of image transfer.
- FLYCAPTURE2_C_API fc2Error fc2StartCapture (fc2Context context)

  Starts isochronous image capture.
- FLYCAPTURE2_C_API fc2Error fc2StartCaptureCallback (fc2Context context, fc2ImageEventCallback pCallbackFn, void ∗pCallbackData)

  Starts isochronous image capture.
- FLYCAPTURE2_C_API fc2Error fc2StartSyncCapture (unsigned int numCameras, fc2Context ∗pContexts)

  Starts synchronized isochronous image capture on multiple cameras.
- FLYCAPTURE2_C_API fc2Error fc2StartSyncCaptureCallback (unsigned int numCameras, fc2Context ∗pContexts, fc2ImageEventCallback ∗pCallbackFns, void ∗∗pCallbackDataArray)

  Starts synchronized isochronous image capture on multiple cameras.
- FLYCAPTURE2_C_API fc2Error fc2RetrieveBuffer (fc2Context context, fc2Image ∗pImage)

  Retrieves the the next image object containing the next image.
- FLYCAPTURE2_C_API fc2Error fc2StopCapture (fc2Context context)

  Stops isochronous image transfer and cleans up all associated resources.
- FLYCAPTURE2_C_API fc2Error fc2SetUserBuffers (fc2Context context, unsigned char ∗const ppMemBuffers, int size, int nNumBuffers)

  Specify user allocated buffers to use as image data buffers.
- FLYCAPTURE2_C_API fc2Error fc2GetConfiguration (fc2Context context, fc2Config ∗config)

  Get the configuration associated with the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetConfiguration (fc2Context context, fc2Config ∗config)

  Set the configuration associated with the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetCameraInfo (fc2Context context, fc2CameraInfo ∗pCameraInfo)

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

- FLYCAPTURE2_C_API fc2Error fc2GetPropertyInfo (fc2Context context, fc2-PropertyInfo ∗propInfo)

  Retrieves information about the specified camera property.
- FLYCAPTURE2_C_API fc2Error fc2GetProperty (fc2Context context, fc2-Property ∗prop)

  Reads the settings for the specified property from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetProperty (fc2Context context, fc2-Property ∗prop)

  Writes the settings for the specified property to the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetPropertyBroadcast (fc2Context context, fc2Property ∗prop)

  Writes the settings for the specified property to the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetGPIOPinDirection (fc2Context context, unsigned int pin, unsigned int ∗pDirection)

  Get the GPIO pin direction for the specified pin.
- FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirection (fc2Context context, unsigned int pin, unsigned int direction)

  Set the GPIO pin direction for the specified pin.
- FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirectionBroadcast (fc2Context context, unsigned int pin, unsigned int direction)

  Set the GPIO pin direction for the specified pin.
- FLYCAPTURE2_C_API fc2Error fc2GetTriggerModeInfo (fc2Context context, fc2TriggerModeInfo ∗triggerModeInfo)

  Retrieve trigger information from the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetTriggerMode (fc2Context context, fc2-TriggerMode ∗triggerMode)

  Retrieve current trigger settings from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetTriggerMode (fc2Context context, fc2-TriggerMode ∗triggerMode)

  Set the specified trigger settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetTriggerModeBroadcast (fc2Context context, fc2TriggerMode ∗triggerMode)

  Set the specified trigger settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTrigger (fc2Context context)
- FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTriggerBroadcast (fc2Context context)

  Fire the software trigger according to the DCAM specifications.
- FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelayInfo (fc2Context context, fc2TriggerDelayInfo ∗triggerDelayInfo)

  Retrieve trigger delay information from the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelay (fc2Context context, fc2-TriggerDelay ∗triggerDelay)

  Retrieve current trigger delay settings from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelay (fc2Context context, fc2-TriggerDelay ∗triggerDelay)

  Set the specified trigger delay settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelayBroadcast (fc2Context context, fc2TriggerDelay ∗triggerDelay)

  Set the specified trigger delay settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetStrobeInfo (fc2Context context, fc2-StrobeInfo ∗strobeInfo)

  Retrieve strobe information from the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetStrobe (fc2Context context, fc2Strobe-Control ∗strobeControl)

  Retrieve current strobe settings from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetStrobe (fc2Context context, fc2Strobe-Control ∗strobeControl)

  Set current strobe settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetStrobeBroadcast (fc2Context context, fc2StrobeControl ∗strobeControl)

  Set current strobe settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRateInfo (fc2-Context context, fc2VideoMode videoMode, fc2FrameRate frameRate, BOOL ∗pSupported)

  Query the camera to determine if the specified video mode and frame rate is supported.
- FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRate (fc2Context context, fc2VideoMode ∗videoMode, fc2FrameRate ∗frameRate)

  Get the current video mode and frame rate from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetVideoModeAndFrameRate (fc2Context context, fc2VideoMode videoMode, fc2FrameRate frameRate)

  Set the specified video mode and frame rate to the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetFormat7Info (fc2Context context, fc2-Format7Info ∗info, BOOL ∗pSupported)

  Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.
- FLYCAPTURE2_C_API fc2Error fc2ValidateFormat7Settings (fc2Context context, fc2Format7ImageSettings ∗imageSettings, BOOL ∗settingsAreValid, fc2-Format7PacketInfo ∗packetInfo)

  Validates Format7ImageSettings structure and returns valid packet size information if the image settings are valid.
- FLYCAPTURE2_C_API fc2Error fc2GetFormat7Configuration (fc2Context context, fc2Format7ImageSettings ∗imageSettings, unsigned int ∗packetSize, float ∗percentage)

  Get the current Format7 configuration from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetFormat7ConfigurationPacket (fc2Context context, fc2Format7ImageSettings ∗imageSettings, unsigned int packetSize)

  Set the current Format7 configuration to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetFormat7Configuration (fc2Context context, fc2Format7ImageSettings ∗imageSettings, float percentSpeed)

  Set the current Format7 configuration to the camera.
- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegister (fc2Context context, unsigned int address, unsigned int value)

  Write a GVCP register.
- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBroadcast (fc2Context context, unsigned int address, unsigned int value)

  Write a GVCP register with broadcast.
- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegister (fc2Context context, unsigned int address, unsigned int ∗pValue)

  Read a GVCP register.
- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBlock (fc2Context context, unsigned int address, const unsigned int ∗pBuffer, unsigned int length)

  Write a GVCP register block.
- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegisterBlock (fc2Context context, unsigned int address, unsigned int ∗pBuffer, unsigned int length)

  Read a GVCP register block.
- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPMemory (fc2Context context, unsigned int address, const unsigned char ∗pBuffer, unsigned int length)

  Write a GVCP memory block.
- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPMemory (fc2Context context, unsigned int address, unsigned char ∗pBuffer, unsigned int length)

  Read a GVCP memory block.
- FLYCAPTURE2_C_API fc2Error fc2GetGigEProperty (fc2Context context, fc2GigEProperty ∗pGigEProp)

  Get the specified GigEProperty.
- FLYCAPTURE2_C_API fc2Error fc2SetGigEProperty (fc2Context context, const fc2GigEProperty ∗pGigEProp)

  Set the specified GigEProperty.
- FLYCAPTURE2_C_API fc2Error fc2QueryGigEImagingMode (fc2Context context, fc2Mode mode, BOOL ∗isSupported)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImagingMode (fc2Context context, fc2Mode ∗mode)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImagingMode (fc2Context context, fc2Mode mode)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettingsInfo (fc2Context context, fc2GigEImageSettingsInfo ∗pInfo)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettings (fc2Context context, fc2GigEImageSettings ∗pImageSettings)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImageSettings (fc2Context context, const fc2GigEImageSettings ∗pImageSettings)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEConfig (fc2Context context, fc2GigEConfig ∗pConfig)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEConfig (fc2Context context, const fc2GigEConfig ∗pConfig)

- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageBinningSettings (fc2Context context, unsigned int ∗horzBinnningValue, unsigned int ∗vertBinnningValue)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImageBinningSettings (fc2Context context, unsigned int horzBinnningValue, unsigned int vertBinnningValue)
- FLYCAPTURE2_C_API fc2Error fc2GetNumStreamChannels (fc2Context context, unsigned int ∗numChannels)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEStreamChannelInfo (fc2Context context, unsigned int channel, fc2GigEStreamChannel ∗pChannel)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEStreamChannelInfo (fc2Context context, unsigned int channel, fc2GigEStreamChannel ∗pChannel)
- FLYCAPTURE2_C_API fc2Error fc2GetLUTInfo (fc2Context context, fc2LUTData ∗pData)

  Query if LUT support is available on the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetLUTBankInfo (fc2Context context, unsigned int bank, BOOL ∗pReadSupported, BOOL ∗pWriteSupported)

  Query the read/write status of a single LUT bank.
- FLYCAPTURE2_C_API fc2Error fc2GetActiveLUTBank (fc2Context context, unsigned int ∗pActiveBank)

  Get the LUT bank that is currently being used.
- FLYCAPTURE2_C_API fc2Error fc2SetActiveLUTBank (fc2Context context, unsigned int activeBank)

  Set the LUT bank that will be used.
- FLYCAPTURE2_C_API fc2Error fc2EnableLUT (fc2Context context, BOOL on)

  Enable or disable LUT functionality on the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetLUTChannel (fc2Context context, unsigned int bank, unsigned int channel, unsigned int sizeEntries, unsigned int ∗pEntries)

  Get the LUT channel settings from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SetLUTChannel (fc2Context context, unsigned int bank, unsigned int channel, unsigned int sizeEntries, unsigned int ∗pEntries)

  Set the LUT channel settings to the camera.
- FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannel (fc2Context context, unsigned int ∗pCurrentChannel)

  Retrieve the current memory channel from the camera.
- FLYCAPTURE2_C_API fc2Error fc2SaveToMemoryChannel (fc2Context context, unsigned int channel)

  Save the current settings to the specfied current memory channel.
- FLYCAPTURE2_C_API fc2Error fc2RestoreFromMemoryChannel (fc2Context context, unsigned int channel)

  Restore the specfied current memory channel.
- FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannelInfo (fc2Context context, unsigned int ∗pNumChannels)

  Query the camera for memory channel support.
- FLYCAPTURE2_C_API fc2Error fc2GetEmbeddedImageInfo (fc2Context context, fc2EmbeddedImageInfo ∗pInfo)

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

- FLYCAPTURE2_C_API fc2Error fc2SetEmbeddedImageInfo (fc2Context context, fc2EmbeddedImageInfo ∗pInfo)

  Sets the on/off values of the embedded image information structure to the camera.
- FLYCAPTURE2_C_API const char ∗ fc2GetRegisterString (unsigned int registerVal)

  Returns a text representation of the register value.
- FLYCAPTURE2_C_API fc2Error fc2CreateImage (fc2Image ∗pImage)

  Create a fc2Image.
- FLYCAPTURE2_C_API fc2Error fc2DestroyImage (fc2Image ∗image)

  Destroy the fc2Image.
- FLYCAPTURE2_C_API fc2Error fc2SetDefaultColorProcessing (fc2ColorProcessingAlgorithm defaultMethod)

  Set the default color processing algorithm.
- FLYCAPTURE2_C_API fc2Error fc2GetDefaultColorProcessing (fc2ColorProcessingAlgorithm ∗pDefaultMethod)

  Get the default color processing algorithm.
- FLYCAPTURE2_C_API fc2Error fc2SetDefaultOutputFormat (fc2PixelFormat format)

  Set the default output pixel format.
- FLYCAPTURE2_C_API fc2Error fc2GetDefaultOutputFormat (fc2PixelFormat ∗pFormat)

  Get the default output pixel format.
- FLYCAPTURE2_C_API fc2Error fc2DetermineBitsPerPixel (fc2PixelFormat format, unsigned int ∗pBitsPerPixel)

  Calculate the bits per pixel for the specified pixel format.
- FLYCAPTURE2_C_API fc2Error fc2SaveImage (fc2Image ∗pImage, const char ∗pFilename, fc2ImageFileFormat format)

  Save the image to the specified file name with the file format specified.
- FLYCAPTURE2_C_API fc2Error fc2SaveImageWithOption (fc2Image ∗pImage, const char ∗pFilename, fc2ImageFileFormat format, void ∗pOption)

  Save the image to the specified file name with the file format specified.
- FLYCAPTURE2_C_API fc2Error fc2ConvertImage (fc2Image ∗pImageIn, fc2Image ∗pImageOut)
- FLYCAPTURE2_C_API fc2Error fc2ConvertImageTo (fc2PixelFormat format, fc2Image ∗pImageIn, fc2Image ∗pImageOut)

  Converts the current image buffer to the specified output format and stores the result in the specified image.
- FLYCAPTURE2_C_API fc2Error fc2GetImageData (fc2Image ∗pImage, unsigned char ∗∗ppData)

  Get a pointer to the data associated with the image.
- FLYCAPTURE2_C_API fc2Error fc2SetImageData (fc2Image ∗pImage, const unsigned char ∗pData, unsigned int dataSize)

  Set the data of the Image object.

- FLYCAPTURE2_C_API fc2Error fc2SetImageDimensions (fc2Image ∗pImage, unsigned int rows, unsigned int cols, unsigned int stride, fc2PixelFormat pixelFormat, fc2BayerTileFormat bayerFormat)

  Sets the dimensions of the image object.
- FLYCAPTURE2_C_API fc2TimeStamp fc2GetImageTimeStamp (fc2Image ∗pImage)

  Get the timestamp data associated with the image.
- FLYCAPTURE2_C_API fc2Error fc2CalculateImageStatistics (fc2Image ∗pImage, fc2ImageStatisticsContext ∗pImageStatisticsContext)

  Calculate statistics associated with the image.
- FLYCAPTURE2_C_API fc2Error fc2CreateImageStatistics (fc2ImageStatisticsContext ∗pImageStatisticsContext)

  Create a statistics context.
- FLYCAPTURE2_C_API fc2Error fc2DestroyImageStatistics (fc2ImageStatisticsContext imageStatisticsContext)

  Destroy a statistics context.
- FLYCAPTURE2_C_API const fc2Error fc2GetChannelStatus (fc2ImageStatisticsContext imageStatisticsContext, fc2StatisticsChannel channel, BOOL ∗pEnabled)

  Get the status of a statistics channel.
- FLYCAPTURE2_C_API fc2Error fc2SetChannelStatus (fc2ImageStatisticsContext imageStatisticsContext, fc2StatisticsChannel channel, BOOL enabled)

  Set the status of a statistics channel.
- FLYCAPTURE2_C_API fc2Error fc2GetImageStatistics (fc2ImageStatisticsContext imageStatisticsContext, fc2StatisticsChannel channel, unsigned int ∗pRangeMin, unsigned int ∗pRangeMax, unsigned int ∗pPixelValueMin, unsigned int ∗pPixelValueMax, unsigned int ∗pNumPixelValues, float ∗pPixelValueMean, int ∗∗ppHistogram)

  Get all statistics for the image.
- FLYCAPTURE2_C_API fc2Error fc2CreateAVI (fc2AVIContext ∗pAVIContext)

  Create a AVI context.
- FLYCAPTURE2_C_API fc2Error fc2AVIOpen (fc2AVIContext AVIContext, const char ∗pFileName, fc2AVIOption ∗pOption)

  Open an AVI file in preparation for writing Images to disk.
- FLYCAPTURE2_C_API fc2Error fc2MJPGOpen (fc2AVIContext AVIContext, const char ∗pFileName, fc2MJPGOption ∗pOption)

  Open an MJPEG file in preparation for writing Images to disk.
- FLYCAPTURE2_C_API fc2Error fc2H264Open (fc2AVIContext AVIContext, const char ∗pFileName, fc2H264Option ∗pOption)

  Open an H.264 file in preparation for writing Images to disk.
- FLYCAPTURE2_C_API fc2Error fc2AVIAppend (fc2AVIContext AVIContext, fc2Image ∗pImage)

  Append an image to the AVI file.
- FLYCAPTURE2_C_API fc2Error fc2AVIClose (fc2AVIContext AVIContext)

  Close the AVI file.

- FLYCAPTURE2_C_API fc2Error fc2DestroyAVI (fc2AVIContext AVIContext)

    Destroy a AVI context.
- FLYCAPTURE2_C_API fc2Error fc2GetSystemInfo (fc2SystemInfo ∗pSystemInfo)

    Get system information.
- FLYCAPTURE2_C_API fc2Error fc2GetLibraryVersion (fc2Version ∗pVersion)

    Get library version.
- FLYCAPTURE2_C_API fc2Error fc2LaunchBrowser (const char ∗pAddress)

    Launch a URL in the system default browser.
- FLYCAPTURE2_C_API fc2Error fc2LaunchHelp (const char ∗pFileName)

    Open a CHM file in the system default CHM viewer.
- FLYCAPTURE2_C_API fc2Error fc2LaunchCommand (const char ∗pCommand)

    Execute a command in the terminal.
- FLYCAPTURE2_C_API fc2Error fc2LaunchCommandAsync (const char ∗pCommand, fc2AsyncCommandCallback pCallback, void ∗pUserData)

    Execute a command in the terminal.
- FLYCAPTURE2_C_API const char ∗ fc2ErrorToDescription (fc2Error error)

    Get a string representation of an error.
- FLYCAPTURE2_C_API fc2Error fc2GetCycleTime (fc2Context context, fc2TimeStamp ∗pTimeStamp)

    Get cycle time from camera.

5.1.1 Function Documentation

5.1.1.1 FLYCAPTURE2_C_API fc2Error fc2AVIAppend (fc2AVIContext *AVIContext*, fc2Image ∗ *pImage*)

Append an image to the AVI file.

Parameters

AVIContext	The AVI context to use.
pImage	The image to append.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.2 FLYCAPTURE2_C_API fc2Error fc2AVIClose (fc2AVIContext *AVIContext*)

Close the AVI file.

Parameters

AVIContext	The AVI context to use.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.3 FLYCAPTURE2_C_API fc2Error fc2AVIOpen (fc2AVIContext *AVIContext*, const char ∗ *pFileName*, fc2AVIOption ∗ *pOption*)

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

AVIContext	The AVI context to use.
pFileName	The filename of the AVI file.
pOption	Options to apply to the AVI file.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.4 FLYCAPTURE2_C_API fc2Error fc2CalculateImageStatistics (fc2Image ∗ *pImage*, fc2ImageStatisticsContext ∗ *pImageStatisticsContext*)

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

pImage	The fc2Image to be used.
pImage-Statistics-Context	The fc2ImageStatisticsContext to hold the statistics.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.5 FLYCAPTURE2_C_API **fc2Error** fc2Connect (**fc2Context** *context,* **fc2PGRGuid** ∗ *guid*)

Connects the camera object to the camera specified by the GUID.

Parameters

context	The fc2Context to be used.
guid	The unique identifier for a specific camera on the PC.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.6 FLYCAPTURE2_C_API **fc2Error** fc2ConvertImage (**fc2Image** ∗ *pImageIn,* **fc2Image** ∗ *pImageOut*)

Parameters

pImageIn	
pImageOut	

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.7 FLYCAPTURE2_C_API **fc2Error** fc2ConvertImageTo (**fc2PixelFormat** *format,* **fc2Image** ∗ *pImageIn,* **fc2Image** ∗ *pImageOut*)

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

format	Output format of the converted image.
pImageIn	Input image.
pImageOut	Output image.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.8 FLYCAPTURE2_C_API fc2Error fc2CreateAVI (fc2AVIContext * *pAVIContext*)

Create a AVI context.

Parameters

pAVIContext	A AVI context.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.9 FLYCAPTURE2_C_API fc2Error fc2CreateContext (fc2Context * *pContext*)

Create a FC2 context for IIDC camaera.

This call must be made before any other calls that use a context will succeed.

Parameters

pContext	A pointer to the fc2Context to be created.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.10 FLYCAPTURE2_C_API fc2Error fc2CreateGigEContext (fc2Context * *pContext*)

Create a FC2 context for a GigE Vision camera.

This call must be made before any other calls that use a context will succeed.

Parameters

pContext	A pointer to the fc2Context to be created.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.11 FLYCAPTURE2_C_API fc2Error fc2CreateImage (fc2Image * *pImage*)

Create a fc2Image.

If externally allocated memory is to be used for the converted image, simply assigning the pData member of the fc2Image structure is insufficient. fc2SetImageData() should be called in order to populate the fc2Image structure correctly.

Parameters

pImage	Pointer to image to be created.

See also

fc2SetImageData()

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.12 FLYCAPTURE2_C_API fc2Error fc2CreateImageStatistics (fc2ImageStatisticsContext * *pImageStatisticsContext*)

Create a statistics context.

Parameters

pImage-Statistics-Context	A statistics context.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.13 FLYCAPTURE2_C_API fc2Error fc2DestroyAVI (fc2AVIContext *AVIContext*)

Destroy a AVI context.

Parameters

AVIContext	A AVI context.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.14 FLYCAPTURE2_C_API fc2Error fc2DestroyContext (fc2Context *context*)

Destroy the FC2 context.

This must be called when the user is finished with the context in order to prevent memory leaks.

Parameters

context	The context to be destroyed.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.15 FLYCAPTURE2_C_API fc2Error fc2DestroyImage (fc2Image ∗ *image*)

Destroy the fc2Image.

Parameters

image	Pointer to image to be destroyed.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.16 FLYCAPTURE2_C_API fc2Error fc2DestroyImageStatistics (fc2ImageStatisticsContext *imageStatisticsContext*)

Destroy a statistics context.

Parameters

image-Statistics-Context	A statistics context.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.17 FLYCAPTURE2_C_API fc2Error fc2DetermineBitsPerPixel (fc2PixelFormat *format*, unsigned int ∗ *pBitsPerPixel*)

Calculate the bits per pixel for the specified pixel format.

Parameters

format	The pixel format.
pBitsPer-Pixel	The bits per pixel.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.18 FLYCAPTURE2_C_API fc2Error fc2Disconnect (fc2Context *context*)

Disconnects the fc2Context from the camera.

Parameters

context	The fc2Context to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.19 FLYCAPTURE2_C_API fc2Error fc2DiscoverGigECameras (fc2Context *context,* fc2CameraInfo ∗ *gigECameras,* unsigned int ∗ *arraySize*)

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.

Parameters

context	The fc2Context to be used.
gigE-Cameras	Pointer to an array of CameraInfo structures.
arraySize	Size of the array. Number of discovered cameras is returned in the same value.

Returns

An Error indicating the success or failure of the function. If the error is PGRERR-OR_BUFFER_TOO_SMALL then arraySize will contain the minimum size needed for gigECameras array.

5.1.1.20 FLYCAPTURE2_C_API fc2Error fc2EnableLUT (fc2Context *context,* BOOL *on*)

Enable or disable LUT functionality on the camera.

Parameters

context	The fc2Context to be used.
on	Whether to enable or disable LUT.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.21 FLYCAPTURE2_C_API const char∗ fc2ErrorToDescription (fc2Error *error*)

Get a string representation of an error.

Parameters

error	Error to be parsed.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.22 FLYCAPTURE2_C_API fc2Error fc2FireBusReset (fc2Context *context*, fc2PGRGuid ∗ *pGuid*)

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

context	The fc2Context to be used.
pGuid	PGRGuid of the camera or the device to cause bus reset.

Returns

An Error indicating the success or failure of the function.

5.1.1.23 FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTrigger (fc2Context *context*)

Parameters

context	The fc2Context to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.24 FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTriggerBroadcast (fc2Context *context*)

Fire the software trigger according to the DCAM specifications.

Parameters

context	The fc2Context to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.25 FLYCAPTURE2_C_API fc2Error fc2ForceAllIPAddressesAutomatically ( )

Force all cameras on the network to be assigned sequential IP addresses on the same subnet as the netowrk 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.

Returns

An Error indicating the success or failure of the function.

5.1.1.26 FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressAutomatically (unsigned int *serialNumber*)

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

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

Returns

An Error indicating the success or failure of the function.

5.1.1.27 FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressToCamera (fc2Context *context*, fc2MACAddress *macAddress*, fc2IPAddress *ipAddress*, fc2IPAddress *subnetMask*, fc2IPAddress *defaultGateway*)

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

context	The fc2Context to be used.
macAddress	MAC address of the camera.
ipAddress	IP address to set on the camera.
subnetMask	Subnet mask to set on the camera.
default-Gateway	Default gateway to set on the camera.

Returns

An Error indicating the success or failure of the function.

5.1.1.28 FLYCAPTURE2_C_API fc2Error fc2GetActiveLUTBank (fc2Context *context*, unsigned int ∗ *pActiveBank*)

Get the LUT bank that is currently being used.

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

Parameters

context	The fc2Context to be used.
pActiveBank	The currently active bank.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.29 FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIndex (fc2Context *context*, unsigned int *index*, fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a camera on the PC.

It uniquely identifies the camera specified by the index and is used to identify the camera during a fc2Connect() call.

Parameters

context	The fc2Context to be used.
index	Zero based index of camera.
pGuid	Unique PGRGuid for the camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.30 FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIPAddress (fc2Context *context,* fc2IPAddress *ipAddress,* fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a camera with the specified IPv4 address.

Parameters

context	The fc2Context to be used.
ipAddress	IP address to get GUID for.
pGuid	Unique PGRGuid for the camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.31 FLYCAPTURE2_C_API fc2Error fc2GetCameraFromSerialNumber (fc2Context *context,* unsigned int *serialNumber,* fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid 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 fc2Connect() call.

Parameters

context	The fc2Context to be used.
serial-Number	Serial number of camera.
pGuid	Unique PGRGuid for the camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.32 FLYCAPTURE2_C_API fc2Error fc2GetCameraInfo (fc2Context *context,* fc2CameraInfo ∗ *pCameraInfo*)

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

Parameters

context	The fc2Context to be used.
pCameraInfo	Pointer to the camera information structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.33 FLYCAPTURE2_C_API fc2Error fc2GetCameraSerialNumberFromIndex (fc2Context *context,* unsigned int *index,* unsigned int * *pSerialNumber*)

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

Parameters

context	The fc2Context to be used.
index	Zero based index of desired camera.
pSerial-Number	Serial number of camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.34 FLYCAPTURE2_C_API const fc2Error fc2GetChannelStatus (fc2ImageStatisticsContext *imageStatisticsContext,* fc2StatisticsChannel *channel,* BOOL * *pEnabled*)

Get the status of a statistics channel.

Parameters

image-Statistics-Context	A statistics context.
channel	The statistics channel.
pEnabled	Whether the channel is enabled.

See also

SetChannelStatus()

Returns

An Error indicating the success or failure of the function.

5.1.1.35 FLYCAPTURE2_C_API fc2Error fc2GetConfiguration (fc2Context *context,* fc2Config * *config*)

Get the configuration associated with the camera.

Parameters

context	The fc2Context to be used.
config	Pointer to the configuration structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.36 FLYCAPTURE2_C_API fc2Error fc2GetCycleTime (fc2Context *context*, fc2TimeStamp ∗ *pTimeStamp*)

Get cycle time from camera.

Parameters

Timestamp	struct.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.37 FLYCAPTURE2_C_API fc2Error fc2GetDefaultColorProcessing (fc2ColorProcessingAlgorithm ∗ *pDefaultMethod*)

Get the default color processing algorithm.

Parameters

pDefault-Method	The default color processing algorithm.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.38 FLYCAPTURE2_C_API fc2Error fc2GetDefaultOutputFormat (fc2PixelFormat ∗ *pFormat*)

Get the default output pixel format.

Parameters

pFormat	The default pixel format.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.39 FLYCAPTURE2_C_API fc2Error fc2GetDeviceFromIndex (fc2Context *context*, unsigned int *index*, fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a device.

It uniquely identifies the device specified by the index.

Parameters

context	The fc2Context to be used.
index	Zero based index of device.
pGuid	Unique PGRGuid for the device.

See also

fc2GetNumOfDevices()

Returns

An Error indicating the success or failure of the function.

5.1.1.40 FLYCAPTURE2_C_API fc2Error fc2GetEmbeddedImageInfo (fc2Context *context*, fc2EmbeddedImageInfo ∗ *pInfo*)

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

Parameters

context	The fc2Context to be used.
pInfo	Structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.41 FLYCAPTURE2_C_API fc2Error fc2GetFormat7Configuration (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, unsigned int ∗ *packetSize*, float ∗ *percentage*)

Get the current Format7 configuration from the camera.

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

Parameters

context	The fc2Context to be used.
image-Settings	Current image settings.
packetSize	Current packet size.
percentage	Current packet size as a percentage.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.42 FLYCAPTURE2_C_API fc2Error fc2GetFormat7Info (fc2Context *context,* fc2Format7Info ∗ *info,* BOOL ∗ *pSupported*)

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

context	The fc2Context to be used.
info	Structure to be filled with the capabilities of the specified mode and the current state in the specified mode.
pSupported	Whether the specified mode is supported.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.43 FLYCAPTURE2_C_API fc2Error fc2GetGigEConfig (fc2Context *context,* fc2GigEConfig ∗ *pConfig*)

5.1.1.44 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageBinningSettings (fc2Context *context,* unsigned int ∗ *horzBinnningValue,* unsigned int ∗ *vertBinnningValue*)

5.1.1.45 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettings (fc2Context *context,* fc2GigEImageSettings ∗ *pImageSettings*)

5.1.1.46 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettingsInfo (fc2Context *context,* fc2GigEImageSettingsInfo ∗ *pInfo*)

5.1.1.47 FLYCAPTURE2_C_API fc2Error fc2GetGigEImagingMode (fc2Context *context,* fc2Mode ∗ *mode*)

5.1.1.48 FLYCAPTURE2_C_API fc2Error fc2GetGigEProperty (fc2Context *context,* fc2GigEProperty ∗ *pGigEProp*)

Get the specified GigEProperty.

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

Parameters

context	The fc2Context to be used.
pGigEProp	The GigE property to get.

Returns

An Error indicating the success or failure of the function.

5.1.1.49 FLYCAPTURE2_C_API fc2Error fc2GetGigEStreamChannelInfo (fc2Context *context,* unsigned int *channel,* fc2GigEStreamChannel ∗ *pChannel*)

5.1.1.50 FLYCAPTURE2_C_API fc2Error fc2GetGPIOPinDirection (fc2Context *context,* unsigned int *pin,* unsigned int ∗ *pDirection*)

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

context	The fc2Context to be used.
pin	Pin to get the direction for.
pDirection	Direction of the pin. 0 for input, 1 for output.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.51 FLYCAPTURE2_C_API fc2Error fc2GetImageData (fc2Image ∗ *pImage,* unsigned char ∗∗ *ppData*)

Get a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is resized or released. The pointer may also be invalidated if the Image object is passed to fc2RetrieveBuffer().

Parameters

pImage	The fc2Image to be used.
ppData	A pointer to the image data.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.52 FLYCAPTURE2_C_API **fc2Error** fc2GetImageStatistics (**fc2ImageStatisticsContext** *imageStatisticsContext*, **fc2StatisticsChannel** *channel*, unsigned int * *pRangeMin*, unsigned int * *pRangeMax*, unsigned int * *pPixelValueMin*, unsigned int * *pPixelValueMax*, unsigned int * *pNumPixelValues*, float * *pPixelValueMean*, int ** *ppHistogram*)

Get all statistics for the image.

Parameters

imageStatisticsContext	The statistics context.
channel	The statistics channel.
pRangeMin	The minimum possible value.
pRangeMax	The maximum possible value.
pPixelValueMin	The minimum pixel value.
pPixelValueMax	The maximum pixel value.
pNumPixelValues	The number of unique pixel values.
pPixelValueMean	The mean of the image.
ppHistogram	Pointer to an array containing the histogram.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.53 FLYCAPTURE2_C_API **fc2TimeStamp** fc2GetImageTimeStamp (**fc2Image** * *pImage*)

Get the timestamp data associated with the image.

Parameters

pImage	The fc2Image to be used.

Returns

Timestamp data associated with the image.

5.1.1.54 FLYCAPTURE2_C_API fc2Error fc2GetInterfaceTypeFromGuid (fc2Context *context*, fc2PGRGuid ∗ *pGuid*, fc2InterfaceType ∗ *pInterfaceType*)

Gets the interface type associated with a PGRGuid.

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

Parameters

context	The fc2Context to be used.
pGuid	The PGRGuid to get the interface for.
pInterface-Type	The interface type of the PGRGuid.

Returns

5.1.1.55 FLYCAPTURE2_C_API fc2Error fc2GetLibraryVersion (fc2Version ∗ *pVersion*)

Get library version.

Parameters

pVersion	Structure to receive the library version.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.56 FLYCAPTURE2_C_API fc2Error fc2GetLUTBankInfo (fc2Context *context*, unsigned int *bank*, BOOL ∗ *pReadSupported*, BOOL ∗ *pWriteSupported*)

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

Parameters

context	The fc2Context to be used.
bank	The bank to query.
pRead-Supported	Whether reading from the bank is supported.
pWrite-Supported	Whether writing to the bank is supported.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.57 FLYCAPTURE2_C_API fc2Error fc2GetLUTChannel (fc2Context *context,* unsigned int *bank,* unsigned int *channel,* unsigned int *sizeEntries,* unsigned int ∗ *pEntries*)

Get the LUT channel settings from the camera.

Parameters

context	The fc2Context to be used.
bank	Bank to retrieve.
channel	Channel to retrieve.
sizeEntries	Number of entries in LUT table to read.
pEntries	Array to store LUT entries.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.58 FLYCAPTURE2_C_API fc2Error fc2GetLUTInfo (fc2Context *context,* fc2LUTData ∗ *pData*)

Query if LUT support is available 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.

Parameters

context	The fc2Context to be used.
pData	The LUT structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.59 FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannel (fc2Context *context,* unsigned int ∗ *pCurrentChannel*)

Retrieve the current memory channel from the camera.

Parameters

context	The fc2Context to be used.
pCurrent-Channel	Current memory channel.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.60 FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannelInfo (fc2Context *context*, unsigned int ∗ *pNumChannels*)

Query the camera for memory channel support.

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

Parameters

context	The fc2Context to be used.
pNum-Channels	Number of memory channels supported.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.61 FLYCAPTURE2_C_API fc2Error fc2GetNumOfCameras (fc2Context *context*, unsigned int ∗ *pNumCameras*)

Gets the number of cameras attached to the PC.

Parameters

context	The fc2Context to be used.
pNum-Cameras	Number of cameras detected.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.62 FLYCAPTURE2_C_API fc2Error fc2GetNumOfDevices (fc2Context *context*, unsigned int ∗ *pNumDevices*)

Gets the number of devices.

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

Parameters

context	The fc2Context to be used.
pNum-Devices	The number of devices found.

Returns

An Error indicating the success or failure of the function.

5.1.1.63 FLYCAPTURE2_C_API fc2Error fc2GetNumStreamChannels (fc2Context *context*, unsigned int ∗ *numChannels*)

5.1.1.64 FLYCAPTURE2_C_API fc2Error fc2GetProperty (fc2Context *context*, fc2Property ∗ *prop*)

Reads the settings for the specified property from the camera.

The property type must be specified in the fc2Property structure passed into the function in order for the function to succeed. If auto is on, the integer and abs values returned may not be consistent with each other.

Parameters

context	The fc2Context to be used.
prop	Pointer to the Property structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.65 FLYCAPTURE2_C_API fc2Error fc2GetPropertyInfo (fc2Context *context*, fc2PropertyInfo ∗ *propInfo*)

Retrieves information about the specified camera property.

The property type must be specified in the fc2PropertyInfo structure passed into the function in order for the function to succeed.

Parameters

context	The fc2Context to be used.
propInfo	Pointer to the PropertyInfo structure to be filled.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.66 FLYCAPTURE2_C_API const char∗ fc2GetRegisterString (unsigned int *registerVal*)

Returns a text representation of the register value.

Parameters

registerVal	The register value to query.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.67 FLYCAPTURE2_C_API fc2Error fc2GetStrobe (fc2Context *context,* fc2StrobeControl ∗ *strobeControl*)

Retrieve current strobe settings from the camera.

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

Parameters

context	The fc2Context to be used.
strobe-Control	Structure to receive strobe settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.68 FLYCAPTURE2_C_API fc2Error fc2GetStrobeInfo (fc2Context *context,* fc2StrobeInfo ∗ *strobeInfo*)

Retrieve strobe information from the camera.

Parameters

context	The fc2Context to be used.
strobeInfo	Structure to receive strobe information.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.69 FLYCAPTURE2_C_API fc2Error fc2GetSystemInfo (fc2SystemInfo ∗ *pSystemInfo*)

Get system information.

Parameters

pSystemInfo	Structure to receive system information.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.70 FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelay (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Retrieve current trigger delay settings from the camera.

Parameters

context	The fc2Context to be used.
triggerDelay	Structure to receive trigger delay settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.71 FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelayInfo (fc2Context *context*, fc2TriggerDelayInfo ∗ *triggerDelayInfo*)

Retrieve trigger delay information from the camera.

Parameters

context	The fc2Context to be used.
triggerDelay-Info	Structure to receive trigger delay information.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.72 FLYCAPTURE2_C_API fc2Error fc2GetTriggerMode (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Retrieve current trigger settings from the camera.

Parameters

context	The fc2Context to be used.
triggerMode	Structure to receive trigger mode settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.73 FLYCAPTURE2_C_API fc2Error fc2GetTriggerModeInfo (fc2Context *context,* fc2TriggerModeInfo ∗ *triggerModeInfo*)

Retrieve trigger information from the camera.

Parameters

context	The fc2Context to be used.
triggerMode-Info	Structure to receive trigger information.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.74 FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRate (fc2Context *context,* fc2VideoMode ∗ *videoMode,* fc2FrameRate ∗ *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

context	The fc2Context to be used.
videoMode	Current video mode.
frameRate	Current frame rate.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.75 FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRateInfo (fc2Context *context,* fc2VideoMode *videoMode,* fc2FrameRate *frameRate,* BOOL ∗ *pSupported*)

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

Parameters

context	The fc2Context to be used.
videoMode	Video mode to check.
frameRate	Frame rate to check.
pSupported	Whether the video mode and frame rate is supported.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.76 FLYCAPTURE2_C_API fc2Error fc2H264Open (fc2AVIContext *AVIContext,* const char ∗ *pFileName,* fc2H264Option ∗ *pOption*)

Open an H.264 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

AVIContext	The AVI context to use.
pFileName	The filename of the AVI file.
pOption	Options to apply to the AVI file.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.77 FLYCAPTURE2_C_API fc2Error fc2IsCameraControlable (fc2Context *context,* fc2PGRGuid ∗ *pGuid,* BOOL ∗ *pControlable*)

Query whether a GigE camera is controlable.

Parameters

context	The fc2Context to be used.
pGuid	Unique PGRGuid for the camera.
pControlable	True indicates camera is controllable

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.78 FLYCAPTURE2_C_API fc2Error fc2LaunchBrowser (const char ∗ *pAddress*)

Launch a URL in the system default browser.

Parameters

pAddress	URL to open in browser.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.79 FLYCAPTURE2_C_API fc2Error fc2LaunchCommand (const char ∗ *pCommand*)

Execute a command in the terminal.

This is a blocking call that will return when the command completes.

Parameters

pCommand	Command to execute.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.80 FLYCAPTURE2_C_API fc2Error fc2LaunchCommandAsync (const char ∗ *pCommand,* fc2AsyncCommandCallback *pCallback,* void ∗ *pUserData*)

Execute a command in the terminal.

This is a non-blocking call that will return immediately. The return value of the command can be retrieved in the callback.

Parameters

pCommand	Command to execute.
pCallback	Callback to fire when command is complete.
pUserData	Data pointer to pass to callback.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.81 FLYCAPTURE2_C_API fc2Error fc2LaunchHelp (const char ∗ *pFileName*)

Open a CHM file in the system default CHM viewer.

Parameters

pFileName	Filename of CHM file to open.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.82 FLYCAPTURE2_C_API fc2Error fc2MJPGOpen (fc2AVIContext *AVIContext*, const char ∗ *pFileName*, fc2MJPGOption ∗ *pOption*)

Open an MJPEG 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

AVIContext	The AVI context to use.
pFileName	The filename of the AVI file.
pOption	Options to apply to the AVI file.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.83 FLYCAPTURE2_C_API fc2Error fc2QueryGigEImagingMode (fc2Context *context*, fc2Mode *mode*, BOOL ∗ *isSupported*)

5.1.1.84 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPMemory (fc2Context *context*, unsigned int *address*, unsigned char ∗ *pBuffer*, unsigned int *length*)

Read a GVCP memory block.

Parameters

context	The fc2Context to be used.
address	GVCP address to be read from.
pBuffer	Array containing data to be written.
length	Size of array, in quadlets.

Returns

An Error indicating the success or failure of the function.

5.1.1.85 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegister (fc2Context *context*, unsigned int *address*, unsigned int ∗ *pValue*)

Read a GVCP register.

Parameters

context	The fc2Context to be used.
address	GVCP address to be read from.
pValue	The value that is read.

Returns

An Error indicating the success or failure of the function.

5.1.1.86 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegisterBlock (fc2Context *context*, unsigned int *address*, unsigned int * *pBuffer*, unsigned int *length*)

Read a GVCP register block.

Parameters

context	The fc2Context to be used.
address	GVCP address to be read from.
pBuffer	Array containing data to be written.
length	Size of array, in quadlets.

Returns

An Error indicating the success or failure of the function.

5.1.1.87 FLYCAPTURE2_C_API fc2Error fc2ReadRegister (fc2Context *context*, unsigned int *address*, unsigned int * *pValue*)

Read the specified register from the camera.

Parameters

context	The fc2Context to be used.
address	DCAM address to be read from.
pValue	The value that is read.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.88 FLYCAPTURE2_C_API fc2Error fc2ReadRegisterBlock (fc2Context *context*, unsigned short *addressHigh*, unsigned int *addressLow*, unsigned int * *pBuffer*, unsigned int *length*)

Write to the specified register block on the camera.

Parameters

context	The fc2Context to be used.
addressHigh	Top 16 bits of the 48 bit absolute address to read from.
addressLow	Bottom 32 bits of the 48 bits absolute address to read from.
pBuffer	Array to store read data.
length	Size of array, in quadlets.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.89 FLYCAPTURE2_C_API fc2Error fc2RegisterCallback (fc2Context *context,* fc2BusEventCallback *enumCallback,* fc2BusCallbackType *callbackType,* void ∗ *pParameter,* fc2CallbackHandle ∗ *pCallbackHandle*)

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

Parameters

context	The fc2Context to be used.
enum-Callback	Pointer to function that will receive the callback.
callbackType	Type of callback to register for.
pParameter	Callback parameter to be passed to callback.
pCallback-Handle	Unique callback handle used for unregistering callback.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.90 FLYCAPTURE2_C_API fc2Error fc2RescanBus (fc2Context *context*)

Force a rescan of the buses.

This does not trigger a bus reset. However, any current connections to a Camera object will be invalidated.

Returns

An Error indicating the success or failure of the function.

5.1.1.91 FLYCAPTURE2_C_API fc2Error fc2RestoreFromMemoryChannel (fc2Context *context,* unsigned int *channel*)

Restore the specfied current memory channel.

Parameters

context	The fc2Context to be used.
channel	Memory channel to restore from.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.92 FLYCAPTURE2_C_API fc2Error fc2RetrieveBuffer (fc2Context *context,* fc2Image ∗ *pImage*)

Retrieves the the next image object containing the next image.

Parameters

context	The fc2Context to be used.
pImage	Pointer to fc2Image to store image data.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.93 FLYCAPTURE2_C_API fc2Error fc2SaveImage (fc2Image ∗ *pImage,* const char ∗ *pFilename,* fc2ImageFileFormat *format*)

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

Parameters

pImage	The fc2Image to be used.
pFilename	Filename to save image with.
format	File format to save in.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.94 FLYCAPTURE2_C_API fc2Error fc2SaveImageWithOption (fc2Image ∗ *pImage,* const char ∗ *pFilename,* fc2ImageFileFormat *format,* void ∗ *pOption*)

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

Parameters

pImage	The fc2Image to be used.
pFilename	Filename to save image with.
format	File format to save in.
pOption	Options for saving image.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.95 FLYCAPTURE2_C_API fc2Error fc2SaveToMemoryChannel (fc2Context *context*, unsigned int *channel*)

Save the current settings to the specfied current memory channel.

Parameters

context	The fc2Context to be used.
channel	Memory channel to save to.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.96 FLYCAPTURE2_C_API fc2Error fc2SetActiveLUTBank (fc2Context *context*, unsigned int *activeBank*)

Set the LUT bank that will be used.

Parameters

context	The fc2Context to be used.
activeBank	The bank to be set as active.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.97 FLYCAPTURE2_C_API fc2Error fc2SetCallback (fc2Context *context*, fc2ImageEventCallback *pCallbackFn*, void * *pCallbackData*)

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

To clear the current stored callback data, pass in NULL for both callback arguments.

Parameters

context	The fc2Context to be used.
pCallbackFn	A function to be called when a new image is received.
pCallback-Data	A pointer to data that can be passed to the callback function.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.98 FLYCAPTURE2_C_API fc2Error fc2SetChannelStatus (fc2ImageStatisticsContext *imageStatisticsContext,* fc2StatisticsChannel *channel,* BOOL *enabled*)

Set the status of a statistics channel.

Parameters

imageStatisticsContext	A statistics context.
channel	The statistics channel.
enabled	Whether the channel should be enabled.

See also

GetChannelStatus()

Returns

An Error indicating the success or failure of the function.

5.1.1.99 FLYCAPTURE2_C_API fc2Error fc2SetConfiguration (fc2Context *context,* fc2Config ∗ *config*)

Set the configuration associated with the camera.

Parameters

context	The fc2Context to be used.
config	Pointer to the configuration structure to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.100 FLYCAPTURE2_C_API fc2Error fc2SetDefaultColorProcessing (fc2ColorProcessingAlgorithm *defaultMethod*)

Set the default color processing algorithm.

This method will be used for any image with the DEFAULT algorithm set. The method used is determined at the time of the Convert() call, therefore the most recent execution of this function will take precedence. The default setting is shared within the current process.

Parameters

default-Method	The color processing algorithm to set.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.101 FLYCAPTURE2_C_API fc2Error fc2SetDefaultOutputFormat (fc2PixelFormat *format*)

Set the default output pixel format.

This format will be used for any call to Convert() that does not specify an output format. The format used will be determined at the time of the Convert() call, therefore the most recent execution of this function will take precedence. The default is shared within the current process.

Parameters

format	The output pixel format to set.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.102 FLYCAPTURE2_C_API fc2Error fc2SetEmbeddedImageInfo (fc2Context *context,* fc2EmbeddedImageInfo ∗ *pInfo*)

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

Parameters

context	The fc2Context to be used.
pInfo	Structure to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.103 FLYCAPTURE2_C_API fc2Error fc2SetFormat7Configuration (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, float *percentSpeed*)

Set the current Format7 configuration to the camera.

Parameters

context	The fc2Context to be used.
image-Settings	Image settings to be written to the camera.
percent-Speed	Packet size as a percentage to be written to the camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.104 FLYCAPTURE2_C_API fc2Error fc2SetFormat7ConfigurationPacket (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, unsigned int *packetSize*)

Set the current Format7 configuration to the camera.

Parameters

context	The fc2Context to be used.
image-Settings	Image settings to be written to the camera.
packetSize	Packet size to be written to the camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.105 FLYCAPTURE2_C_API fc2Error fc2SetGigEConfig (fc2Context *context*, const fc2GigEConfig ∗ *pConfig*)

5.1.1.106 FLYCAPTURE2_C_API fc2Error fc2SetGigEImageBinningSettings (fc2Context *context*, unsigned int *horzBinnningValue*, unsigned int *vertBinnningValue*)

5.1.1.107 FLYCAPTURE2_C_API fc2Error fc2SetGigEImageSettings (fc2Context *context*, const fc2GigEImageSettings ∗ *pImageSettings*)

5.1.1.108 FLYCAPTURE2_C_API **fc2Error** fc2SetGigEImagingMode (**fc2Context** *context,* **fc2Mode** *mode*)

5.1.1.109 FLYCAPTURE2_C_API **fc2Error** fc2SetGigEProperty (**fc2Context** *context,* const **fc2GigEProperty** ∗ *pGigEProp*)

Set the specified GigEProperty.

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

Parameters

context	The fc2Context to be used.
pGigEProp	The GigE property to set.

Returns

An Error indicating the success or failure of the function.

5.1.1.110 FLYCAPTURE2_C_API **fc2Error** fc2SetGigEStreamChannelInfo (**fc2Context** *context,* unsigned int *channel,* **fc2GigEStreamChannel** ∗ *pChannel*)

5.1.1.111 FLYCAPTURE2_C_API **fc2Error** fc2SetGPIOPinDirection (**fc2Context** *context,* unsigned int *pin,* unsigned int *direction*)

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

context	The fc2Context to be used.
pin	Pin to get the direction for.
direction	Direction of the pin. 0 for input, 1 for output.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.112 FLYCAPTURE2_C_API **fc2Error** fc2SetGPIOPinDirectionBroadcast (**fc2Context** *context,* unsigned int *pin,* unsigned int *direction*)

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

context	The fc2Context to be used.
pin	Pin to get the direction for.
direction	Direction of the pin. 0 for input, 1 for output.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.113 FLYCAPTURE2_C_API fc2Error fc2SetImageData (fc2Image ∗ *pImage,* const unsigned char ∗ *pData,* unsigned int *dataSize*)

Set the data of the Image object.

Ownership of the image buffer is not transferred to the Image object. It is the user's responsibility to delete the buffer when it is no longer in use.

Parameters

pImage	The fc2Image to be used.
pData	Pointer to the image buffer.
dataSize	Size of the image buffer.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.114 FLYCAPTURE2_C_API fc2Error fc2SetImageDimensions (fc2Image ∗ *pImage,* unsigned int *rows,* unsigned int *cols,* unsigned int *stride,* fc2PixelFormat *pixelFormat,* fc2BayerTileFormat *bayerFormat*)

Sets the dimensions of the image object.

Parameters

pImage	The fc2Image to be used.
rows	Number of rows to set.
cols	Number of cols to set.
stride	Stride to set.
pixelFormat	Pixel format to set.
bayerFormat	Bayer tile format to set.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.115 FLYCAPTURE2_C_API fc2Error fc2SetLUTChannel (fc2Context *context,* unsigned int *bank,* unsigned int *channel,* unsigned int *sizeEntries,* unsigned int ∗ *pEntries*)

Set the LUT channel settings to the camera.

Parameters

context	The fc2Context to be used.
bank	Bank to set.
channel	Channel to set.
sizeEntries	Number of entries in LUT table to write. This must be the same size as numEntries returned by GetLutInfo().
pEntries	Array containing LUT entries to write.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.116 FLYCAPTURE2_C_API fc2Error fc2SetProperty (fc2Context *context,* fc2Property ∗ *prop*)

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

context	The fc2Context to be used.
prop	Pointer to the Property structure to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.117 FLYCAPTURE2_C_API fc2Error fc2SetPropertyBroadcast (fc2Context *context,* fc2Property ∗ *prop*)

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

context	The fc2Context to be used.
prop	Pointer to the Property structure to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.118 FLYCAPTURE2_C_API fc2Error fc2SetStrobe (fc2Context *context*, fc2StrobeControl ∗ *strobeControl*)

Set current strobe settings to the camera.

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

Parameters

context	The fc2Context to be used.
strobe-Control	Structure providing strobe settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.119 FLYCAPTURE2_C_API fc2Error fc2SetStrobeBroadcast (fc2Context *context*, fc2StrobeControl ∗ *strobeControl*)

Set current strobe settings to the camera.

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

Parameters

context	The fc2Context to be used.
strobe-Control	Structure providing strobe settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.120 FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelay (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Set the specified trigger delay settings to the camera.

Parameters

context	The fc2Context to be used.
triggerDelay	Structure providing trigger delay settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.121 FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelayBroadcast (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Set the specified trigger delay settings to the camera.

Parameters

context	The fc2Context to be used.
triggerDelay	Structure providing trigger delay settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.122 FLYCAPTURE2_C_API fc2Error fc2SetTriggerMode (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Set the specified trigger settings to the camera.

Parameters

context	The fc2Context to be used.
triggerMode	Structure providing trigger mode settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.123 FLYCAPTURE2_C_API fc2Error fc2SetTriggerModeBroadcast (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Set the specified trigger settings to the camera.

Parameters

context	The fc2Context to be used.
triggerMode	Structure providing trigger mode settings.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.124 FLYCAPTURE2_C_API fc2Error fc2SetUserBuffers (fc2Context *context*, unsigned char ∗const *ppMemBuffers*, int *size*, int *nNumBuffers*)

Specify user allocated buffers to use as image data buffers.

Parameters

context	The fc2Context to be used.
ppMem-Buffers	Pointer to memory buffers to be written to. The size of the data should be equal to (size ∗ numBuffers) or larger.
size	The size of each buffer (in bytes).
nNum-Buffers	Number of buffers in the array.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.125 FLYCAPTURE2_C_API fc2Error fc2SetVideoModeAndFrameRate (fc2Context *context*, fc2VideoMode *videoMode*, fc2FrameRate *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_FO-RMAT7. Use the Format7 functions to set the camera into Format7.

Parameters

context	The fc2Context to be used.
videoMode	Video mode to set to camera.
frameRate	Frame rate to set to camera.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.126 FLYCAPTURE2_C_API fc2Error fc2StartCapture (fc2Context *context*)

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera.

Parameters

context	The fc2Context to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.127 FLYCAPTURE2_C_API fc2Error fc2StartCaptureCallback (fc2Context *context,* fc2ImageEventCallback *pCallbackFn,* void ∗ *pCallbackData*)

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 is called when a new image is received from the camera.

Parameters

context	The fc2Context to be used.
pCallbackFn	A function to be called when a new image is received.
pCallback-Data	A pointer to data that can be passed to the callback function. A NULL pointer is acceptable.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.128 FLYCAPTURE2_C_API fc2Error fc2StartSyncCapture (unsigned int *numCameras,* fc2Context ∗ *pContexts*)

Starts synchronized isochronous image capture on multiple cameras.

Parameters

num-Cameras	Number of fc2Contexts in the ppCameras array.
pContexts	Array of fc2Contexts.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.129 FLYCAPTURE2_C_API fc2Error fc2StartSyncCaptureCallback (unsigned int *numCameras,* fc2Context ∗ *pContexts,* fc2ImageEventCallback ∗ *pCallbackFns,* void ∗∗ *pCallbackDataArray*)

Starts synchronized isochronous image capture on multiple cameras.

Parameters

numCameras	Number of fc2Contexts in the ppCameras array.
pContexts	Array of fc2Contexts.
pCallbackFns	Array of callback functions for each camera.
pCallbackDataArray	Array of callback data pointers.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.130 FLYCAPTURE2_C_API fc2Error fc2StopCapture (fc2Context *context*)

Stops isochronous image transfer and cleans up all associated resources.

Parameters

context	The fc2Context to be used.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.131 FLYCAPTURE2_C_API fc2Error fc2UnregisterCallback (fc2Context *context*, fc2CallbackHandle *callbackHandle*)

Unregister a callback function.

Parameters

context	The fc2Context to be used.
callbackHandle	Unique callback handle.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.132 FLYCAPTURE2_C_API fc2Error fc2ValidateFormat7Settings (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, BOOL ∗ *settingsAreValid*, fc2Format7PacketInfo ∗ *packetInfo*)

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

context	The fc2Context to be used.
image-Settings	Structure containing the image settings.
settingsAre-Valid	Whether the settings are valid.
packetInfo	Packet size information that can be used to determine a valid packet size.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.133 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPMemory (fc2Context *context,* unsigned int *address,* const unsigned char ∗ *pBuffer,* unsigned int *length*)

Write a GVCP memory block.

Parameters

context	The fc2Context to be used.
address	GVCP address to be write to.
pBuffer	Array containing data to be written.
length	Size of array, in quadlets.

Returns

An Error indicating the success or failure of the function.

5.1.1.134 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegister (fc2Context *context,* unsigned int *address,* unsigned int *value*)

Write a GVCP register.

Parameters

context	The fc2Context to be used.
address	GVCP address to be written to.
value	The value to be written.

Returns

An Error indicating the success or failure of the function.

5.1.1.135 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBlock (fc2Context *context,* unsigned int *address,* const unsigned int ∗ *pBuffer,* unsigned int *length*)

Write a GVCP register block.

Parameters

context	The fc2Context to be used.
address	GVCP address to be write to.
pBuffer	Array containing data to be written.
length	Size of array, in quadlets.

Returns

An Error indicating the success or failure of the function.

5.1.1.136 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBroadcast (fc2Context *context,* unsigned int *address,* unsigned int *value*)

Write a GVCP register with broadcast.

Parameters

context	The fc2Context to be used.
address	GVCP address to be written to.
value	The value to be written.

Returns

An Error indicating the success or failure of the function.

5.1.1.137 FLYCAPTURE2_C_API fc2Error fc2WriteRegister (fc2Context *context,* unsigned int *address,* unsigned int *value*)

Write to the specified register on the camera.

Parameters

context	The fc2Context to be used.
address	DCAM address to be written to.
value	The value to be written.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.138 FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBlock (fc2Context *context*, unsigned short *addressHigh*, unsigned int *addressLow*, const unsigned int ∗ *pBuffer*, unsigned int *length*)

Write to the specified register block on the camera.

Parameters

context	The fc2Context to be used.
addressHigh	Top 16 bits of the 48 bit absolute address to write to.
addressLow	Bottom 32 bits of the 48 bits absolute address to write to.
pBuffer	Array containing data to be written.
length	Size of array, in quadlets.

Returns

A fc2Error indicating the success or failure of the function.

5.1.1.139 FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBroadcast (fc2Context *context*, unsigned int *address*, unsigned int *value*)

Write to the specified register on the camera with broadcast.

Parameters

context	The fc2Context to be used.
address	DCAM address to be written to.
value	The value to be written.

Returns

A fc2Error indicating the success or failure of the function.

5.2 FlyCapture2Defs_C.h File Reference

Data Structures

- struct fc2PGRGuid

  A GUID to the camera.
- struct fc2Image
- struct fc2SystemInfo

- struct fc2Version
- struct fc2Config
- struct fc2TriggerDelayInfo
- struct fc2TriggerDelay
- struct fc2TriggerModeInfo
- struct fc2TriggerMode
- struct fc2StrobeInfo
- struct fc2StrobeControl
- struct fc2Format7ImageSettings
- struct fc2Format7Info
- struct fc2Format7PacketInfo
- struct fc2IPAddress
- struct fc2MACAddress
- struct fc2GigEProperty
- struct fc2GigEStreamChannel
- struct fc2GigEConfig
- struct fc2GigEImageSettingsInfo
- struct fc2GigEImageSettings
- struct fc2TimeStamp
- struct fc2ConfigROM
- struct fc2CameraInfo
- struct fc2EmbeddedImageInfoProperty
- struct fc2EmbeddedImageInfo
- struct fc2ImageMetadata
- struct fc2LUTData
- struct fc2PNGOption
- struct fc2PPMOption
- struct fc2PGMOption
- struct fc2TIFFOption
- struct fc2JPEGOption
- struct fc2JPG2Option
- struct fc2AVIOption
- struct fc2MJPGOption
- struct fc2H264Option

Defines

- #define FALSE 0
- #define TRUE 1
- #define FULL_32BIT_VALUE 0x7FFFFFFF
- #define MAX_STRING_LENGTH 512

Typedefs

- typedef int BOOL
- typedef void ∗ fc2Context

  A context to the FlyCapture2 C library.
- typedef void ∗ fc2GuiContext

  A context to the FlyCapture2 C GUI library.
- typedef void ∗ fc2ImageImpl

  An internal pointer used in the fc2Image structure.
- typedef void ∗ fc2AVIContext

  A context referring to the AVI recorder object.
- typedef void ∗ fc2ImageStatisticsContext

  A context referring to the ImageStatistics object.
- typedef void ∗ fc2CallbackHandle
- typedef void(∗ fc2BusEventCallback)(void ∗pParameter, unsigned int serial-Number)
- typedef void(∗ fc2ImageEventCallback)(fc2Image ∗image, void ∗pCallbackData)
- typedef void(∗ fc2AsyncCommandCallback)(fc2Error retError, void ∗pUserData)

Enumerations

- enum fc2Error { FC2_ERROR_UNDEFINED = -1, FC2_ERROR_OK, FC2-_ERROR_FAILED, FC2_ERROR_NOT_IMPLEMENTED, FC2_ERROR_FAIL-ED_BUS_MASTER_CONNECTION, FC2_ERROR_NOT_CONNECTED, FC2-_ERROR_INIT_FAILED, FC2_ERROR_NOT_INTITIALIZED, FC2_ERROR_I-NVALID_PARAMETER, FC2_ERROR_INVALID_SETTINGS, FC2_ERROR_-INVALID_BUS_MANAGER, FC2_ERROR_MEMORY_ALLOCATION_FAILED, FC2_ERROR_LOW_LEVEL_FAILURE, FC2_ERROR_NOT_FOUND, FC2_-ERROR_FAILED_GUID, FC2_ERROR_INVALID_PACKET_SIZE, FC2_ERR-OR_INVALID_MODE, FC2_ERROR_NOT_IN_FORMAT7, FC2_ERROR_NO-T_SUPPORTED, FC2_ERROR_TIMEOUT, FC2_ERROR_BUS_MASTER_F-AILED, FC2_ERROR_INVALID_GENERATION, FC2_ERROR_LUT_FAILED, FC2_ERROR_IIDC_FAILED, FC2_ERROR_STROBE_FAILED, FC2_ERRO-R_TRIGGER_FAILED, FC2_ERROR_PROPERTY_FAILED, FC2_ERROR_P-ROPERTY_NOT_PRESENT, FC2_ERROR_REGISTER_FAILED, FC2_ERR-OR_READ_REGISTER_FAILED, FC2_ERROR_WRITE_REGISTER_FAILED, FC2_ERROR_ISOCH_FAILED, FC2_ERROR_ISOCH_ALREADY_STARTED, FC2_ERROR_ISOCH_NOT_STARTED, FC2_ERROR_ISOCH_START_FAIL-ED, FC2_ERROR_ISOCH_RETRIEVE_BUFFER_FAILED, FC2_ERROR_ISO-CH_STOP_FAILED, FC2_ERROR_ISOCH_SYNC_FAILED, FC2_ERROR_IS-OCH_BANDWIDTH_EXCEEDED, FC2_ERROR_IMAGE_CONVERSION_FAI-LED, FC2_ERROR_IMAGE_LIBRARY_FAILURE, FC2_ERROR_BUFFER_T-OO_SMALL, FC2_ERROR_IMAGE_CONSISTENCY_ERROR, FC2_ERROR-_INCOMPATIBLE_DRIVER, FC2_ERROR_FORCE_32BITS = FULL_32BIT_V-ALUE }
- enum fc2BusCallbackType { FC2_BUS_RESET, FC2_ARRIVAL, FC2_REMO-VAL, FC2_CALLBACK_TYPE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2GrabMode { FC2_DROP_FRAMES, FC2_BUFFER_FRAMES, FC2_-UNSPECIFIED_GRAB_MODE, FC2_GRAB_MODE_FORCE_32BITS = FULL-_32BIT_VALUE }
- enum fc2GrabTimeout { FC2_TIMEOUT_NONE = 0, FC2_TIMEOUT_INFINITE = -1, FC2_TIMEOUT_UNSPECIFIED = -2, FC2_GRAB_TIMEOUT_FORCE_-32BITS = FULL_32BIT_VALUE }
- enum fc2BandwidthAllocation { FC2_BANDWIDTH_ALLOCATION_OFF = 0, FC2_BANDWIDTH_ALLOCATION_ON = 1, FC2_BANDWIDTH_ALLOCATIO-N_UNSUPPORTED = 2, FC2_BANDWIDTH_ALLOCATION_UNSPECIFIED = 3, FC2_BANDWIDTH_ALLOCATION_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2InterfaceType { FC2_INTERFACE_IEEE1394, FC2_INTERFACE_US-B_2, FC2_INTERFACE_USB_3, FC2_INTERFACE_GIGE, FC2_INTERFAC-E_UNKNOWN, FC2_INTERFACE_TYPE_FORCE_32BITS = FULL_32BIT_VA-LUE }
- enum fc2DriverType { FC2_DRIVER_1394_CAM, FC2_DRIVER_1394_PRO, FC2_DRIVER_1394_JUJU, FC2_DRIVER_1394_VIDEO1394, FC2_DRIVE-R_1394_RAW1394, FC2_DRIVER_USB_NONE, FC2_DRIVER_USB_CAM, FC2_DRIVER_USB3_PRO, FC2_DRIVER_GIGE_NONE, FC2_DRIVER_GIG-E_FILTER, FC2_DRIVER_GIGE_PRO, FC2_DRIVER_UNKNOWN = -1, FC2-_DRIVER_FORCE_32BITS = FULL_32BIT_VALUE }

  Types of low level drivers that flycapture uses.
- enum fc2PropertyType { FC2_BRIGHTNESS, FC2_AUTO_EXPOSURE, FC2-_SHARPNESS, FC2_WHITE_BALANCE, FC2_HUE, FC2_SATURATION, F-C2_GAMMA, FC2_IRIS, FC2_FOCUS, FC2_ZOOM, FC2_PAN, FC2_TILT, FC2_SHUTTER, FC2_GAIN, FC2_TRIGGER_MODE, FC2_TRIGGER_DELA-Y, FC2_FRAME_RATE, FC2_TEMPERATURE, FC2_UNSPECIFIED_PROPE-RTY_TYPE, FC2_PROPERTY_TYPE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2FrameRate { FC2_FRAMERATE_1_875, FC2_FRAMERATE_3_75, FC2_FRAMERATE_7_5, FC2_FRAMERATE_15, FC2_FRAMERATE_30, F-C2_FRAMERATE_60, FC2_FRAMERATE_120, FC2_FRAMERATE_240, F-C2_FRAMERATE_FORMAT7, FC2_NUM_FRAMERATES, FC2_FRAMERAT-E_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2VideoMode { FC2_VIDEOMODE_160x120YUV444, FC2_VIDEOMO-DE_320x240YUV422, FC2_VIDEOMODE_640x480YUV411, FC2_VIDEOMO-DE_640x480YUV422, FC2_VIDEOMODE_640x480RGB, FC2_VIDEOMODE-_640x480Y8, FC2_VIDEOMODE_640x480Y16, FC2_VIDEOMODE_800x600-YUV422, FC2_VIDEOMODE_800x600RGB, FC2_VIDEOMODE_800x600Y8, FC2_VIDEOMODE_800x600Y16, FC2_VIDEOMODE_1024x768YUV422, F-C2_VIDEOMODE_1024x768RGB, FC2_VIDEOMODE_1024x768Y8, FC2_VI-DEOMODE_1024x768Y16, FC2_VIDEOMODE_1280x960YUV422, FC2_VID-EOMODE_1280x960RGB, FC2_VIDEOMODE_1280x960Y8, FC2_VIDEOMO-DE_1280x960Y16, FC2_VIDEOMODE_1600x1200YUV422, FC2_VIDEOMO-DE_1600x1200RGB, FC2_VIDEOMODE_1600x1200Y8, FC2_VIDEOMODE-_1600x1200Y16, FC2_VIDEOMODE_FORMAT7, FC2_NUM_VIDEOMODES, FC2_VIDEOMODE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2Mode { FC2_MODE_0 = 0, FC2_MODE_1, FC2_MODE_2, FC2_-MODE_3, FC2_MODE_4, FC2_MODE_5, FC2_MODE_6, FC2_MODE_7,

FC2_MODE_8, FC2_MODE_9, FC2_MODE_10, FC2_MODE_11, FC2_MODE_12, FC2_MODE_13, FC2_MODE_14, FC2_MODE_15, FC2_MODE_16, FC2_MODE_17, FC2_MODE_18, FC2_MODE_19, FC2_MODE_20, FC2_MODE_21, FC2_MODE_22, FC2_MODE_23, FC2_MODE_24, FC2_MODE_25, FC2_MODE_26, FC2_MODE_27, FC2_MODE_28, FC2_MODE_29, FC2_MODE_30, FC2_MODE_31, FC2_NUM_MODES, FC2_MODE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2PixelFormat { FC2_PIXEL_FORMAT_MONO8 = 0x80000000, FC2_PIXEL_FORMAT_411YUV8 = 0x40000000, FC2_PIXEL_FORMAT_422YUV8 = 0x20000000, FC2_PIXEL_FORMAT_444YUV8 = 0x10000000, FC2_PIXEL_FORMAT_RGB8 = 0x08000000, FC2_PIXEL_FORMAT_MONO16 = 0x04000000, FC2_PIXEL_FORMAT_RGB16 = 0x02000000, FC2_PIXEL_FORMAT_S_MONO16 = 0x01000000, FC2_PIXEL_FORMAT_S_RGB16 = 0x00800000, FC2_PIXEL_FORMAT_RAW8 = 0x00400000, FC2_PIXEL_FORMAT_RAW16 = 0x00200000, FC2_PIXEL_FORMAT_MONO12 = 0x00100000, FC2_PIXEL_FORMAT_RAW12 = 0x00080000, FC2_PIXEL_FORMAT_BGR = 0x80000008, FC2_PIXEL_FORMAT_BGRU = 0x40000008, FC2_PIXEL_FORMAT_RGB = FC2_PIXEL_FORMAT_RGB8, FC2_PIXEL_FORMAT_RGBU = 0x40000002, FC2_PIXEL_FORMAT_BGR16 = 0x02000001, FC2_PIXEL_FORMAT_BGRU16 = 0x02000002, FC2_PIXEL_FORMAT_422YUV8_JPEG = 0x40000001, FC2_NUM_PIXEL_FORMATS = 20, FC2_UNSPECIFIED_PIXEL_FORMAT = 0 }
- enum fc2BusSpeed { FC2_BUSSPEED_S100, FC2_BUSSPEED_S200, FC2_BUSSPEED_S400, FC2_BUSSPEED_S480, FC2_BUSSPEED_S800, FC2_BUSSPEED_S1600, FC2_BUSSPEED_S3200, FC2_BUSSPEED_S5000, ×
FC2_BUSSPEED_10BASE_T, FC2_BUSSPEED_100BASE_T, FC2_BUSSPEED_1000BASE_T, FC2_BUSSPEED_10000BASE_T, FC2_BUSSPEED_S_FASTEST, FC2_BUSSPEED_ANY, FC2_BUSSPEED_SPEED_UNKNOWN = -1, FC2_BUSSPEED_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2PCIeBusSpeed { FC2_PCIE_BUSSPEED_2_5, FC2_PCIE_BUSSPEED_5_0, FC2_PCIE_BUSSPEED_UNKNOWN = -1, FC2_PCIE_BUSSPEED_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2ColorProcessingAlgorithm { FC2_DEFAULT, FC2_NO_COLOR_PROCESSING, FC2_NEAREST_NEIGHBOR_FAST, FC2_EDGE_SENSING, FC2_HQ_LINEAR, FC2_RIGOROUS, FC2_IPP, FC2_DIRECTIONAL, FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2BayerTileFormat { FC2_BT_NONE, FC2_BT_RGGB, FC2_BT_GRBG, FC2_BT_GBRG, FC2_BT_BGGR, FC2_BT_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2ImageFileFormat { FC2_FROM_FILE_EXT = -1, FC2_PGM, FC2_PPM, FC2_BMP, FC2_JPEG, FC2_JPEG2000, FC2_TIFF, FC2_PNG, FC2_RAW, FC2_IMAGE_FILE_FORMAT_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2GigEPropertyType { FC2_HEARTBEAT, FC2_HEARTBEAT_TIMEOUT }
- enum fc2StatisticsChannel { FC2_STATISTICS_GREY, FC2_STATISTICS_RED, FC2_STATISTICS_GREEN, FC2_STATISTICS_BLUE, FC2_STATISTICS_HUE, FC2_STATISTICS_SATURATION, FC2_STATISTICS_LIGHTNESS, FC2_STATISTICS_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2OSType { FC2_WINDOWS_X86, FC2_WINDOWS_X64, FC2_LINUX_X86, FC2_LINUX_X64, FC2_MAC, FC2_UNKNOWN_OS, FC2_OSTYPE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2ByteOrder { FC2_BYTE_ORDER_LITTLE_ENDIAN, FC2_BYTE_ORDER_BIG_ENDIAN, FC2_BYTE_ORDER_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2TIFFCompressionMethod { FC2_TIFF_NONE = 1, FC2_TIFF_PACKBITS, FC2_TIFF_DEFLATE, FC2_TIFF_ADOBE_DEFLATE, FC2_TIFF_CCITTFAX3, FC2_TIFF_CCITTFAX4, FC2_TIFF_LZW, FC2_TIFF_JPEG }

5.2.1 Define Documentation

5.2.1.1 #define FALSE 0

5.2.1.2 #define FULL_32BIT_VALUE 0x7FFFFFFF

5.2.1.3 #define MAX_STRING_LENGTH 512

5.2.1.4 #define TRUE 1

5.2.2 Typedef Documentation

5.2.2.1 typedef int **BOOL**

5.2.2.2 typedef void(∗ **fc2AsyncCommandCallback**)(**fc2Error** retError, void ∗pUserData)

5.2.2.3 typedef void∗ **fc2AVIContext**

A context referring to the AVI recorder object.

5.2.2.4 typedef void(∗ **fc2BusEventCallback**)(void ∗pParameter, unsigned int serialNumber)

5.2.2.5 typedef void∗ **fc2CallbackHandle**

5.2.2.6 typedef void∗ **fc2Context**

A context to the FlyCapture2 C library.

It must be created before performing any calls to the library.

5.2.2.7 typedef void∗ **fc2GuiContext**

A context to the FlyCapture2 C GUI library.

It must be created before performing any calls to the library.

5.2.2.8 typedef void(∗ fc2ImageEventCallback)(fc2Image ∗image, void ∗pCallbackData)

5.2.2.9 typedef void∗ fc2ImageImpl

An internal pointer used in the fc2Image structure.

5.2.2.10 typedef void∗ fc2ImageStatisticsContext

A context referring to the ImageStatistics object.

5.2.3 Enumeration Type Documentation

5.2.3.1 enum fc2BandwidthAllocation

Enumerator:

FC2_BANDWIDTH_ALLOCATION_OFF

FC2_BANDWIDTH_ALLOCATION_ON

FC2_BANDWIDTH_ALLOCATION_UNSUPPORTED

FC2_BANDWIDTH_ALLOCATION_UNSPECIFIED

FC2_BANDWIDTH_ALLOCATION_FORCE_32BITS

5.2.3.2 enum fc2BayerTileFormat

Enumerator:

FC2_BT_NONE	No bayer tile format.
FC2_BT_RGGB	Red-Green-Green-Blue.
FC2_BT_GRBG	Green-Red-Blue-Green.
FC2_BT_GBRG	Green-Blue-Red-Green.
FC2_BT_BGGR	Blue-Green-Green-Red.
FC2_BT_FORCE_32BITS	

5.2.3.3 enum fc2BusCallbackType

Enumerator:

FC2_BUS_RESET

FC2_ARRIVAL

FC2_REMOVAL

FC2_CALLBACK_TYPE_FORCE_32BITS

5.2.3.4 enum fc2BusSpeed

Enumerator:

FC2_BUSSPEED_S100	100Mbits/sec.
FC2_BUSSPEED_S200	200Mbits/sec.
FC2_BUSSPEED_S400	400Mbits/sec.
FC2_BUSSPEED_S480	480Mbits/sec. Only for USB2 cameras.
FC2_BUSSPEED_S800	800Mbits/sec.
FC2_BUSSPEED_S1600	1600Mbits/sec.
FC2_BUSSPEED_S3200	3200Mbits/sec.
FC2_BUSSPEED_S5000	5000Mbits/sec. Only for USB3 cameras.
FC2_BUSSPEED_10BASE_T	10Base-T. Only for GigE cameras.
FC2_BUSSPEED_100BASE_T	100Base-T. Only for GigE cameras.
FC2_BUSSPEED_1000BASE_T	1000Base-T (Gigabit Ethernet). Only for GigE cameras.
FC2_BUSSPEED_10000BASE_T	10000Base-T. Only for GigE cameras.
FC2_BUSSPEED_S_FASTEST	The fastest speed available.
FC2_BUSSPEED_ANY	Any speed that is available.
FC2_BUSSPEED_SPEED_UNKNOWN	Unknown bus speed.
FC2_BUSSPEED_FORCE_32BITS	

5.2.3.5 enum fc2ByteOrder

Enumerator:

FC2_BYTE_ORDER_LITTLE_ENDIAN

FC2_BYTE_ORDER_BIG_ENDIAN

FC2_BYTE_ORDER_FORCE_32BITS

5.2.3.6 enum fc2ColorProcessingAlgorithm

Enumerator:

FC2_DEFAULT

FC2_NO_COLOR_PROCESSING

FC2_NEAREST_NEIGHBOR_FAST

FC2_EDGE_SENSING

FC2_HQ_LINEAR

FC2_RIGOROUS

FC2_IPP

FC2_DIRECTIONAL

FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS

5.2.3.7 enum fc2DriverType

Types of low level drivers that flycapture uses.

Enumerator:

- ***FC2_DRIVER_1394_CAM*** PGRCam.sys.
- ***FC2_DRIVER_1394_PRO*** PGR1394.sys.
- ***FC2_DRIVER_1394_JUJU*** firewire_core.
- ***FC2_DRIVER_1394_VIDEO1394*** video1394.
- ***FC2_DRIVER_1394_RAW1394*** raw1394.
- ***FC2_DRIVER_USB_NONE*** No usb driver used just BSD stack. (Linux only)
- ***FC2_DRIVER_USB_CAM*** PGRUsbCam.sys.
- ***FC2_DRIVER_USB3_PRO*** PGRXHCI.sys.
- ***FC2_DRIVER_GIGE_NONE*** no gige drivers used,MS/BSD stack.
- ***FC2_DRIVER_GIGE_FILTER*** PGRGigE.sys.
- ***FC2_DRIVER_GIGE_PRO*** PGRGigEPro.sys.
- ***FC2_DRIVER_UNKNOWN*** Unknown driver type.
- ***FC2_DRIVER_FORCE_32BITS***

5.2.3.8 enum fc2Error

Enumerator:

- ***FC2_ERROR_UNDEFINED*** Undefined.
- ***FC2_ERROR_OK*** Function returned with no errors.
- ***FC2_ERROR_FAILED*** General failure.
- ***FC2_ERROR_NOT_IMPLEMENTED*** Function has not been implemented.
- ***FC2_ERROR_FAILED_BUS_MASTER_CONNECTION*** Could not connect to - Bus Master.
- ***FC2_ERROR_NOT_CONNECTED*** Camera has not been connected.
- ***FC2_ERROR_INIT_FAILED*** Initialization failed.
- ***FC2_ERROR_NOT_INTITIALIZED*** Camera has not been initialized.
- ***FC2_ERROR_INVALID_PARAMETER*** Invalid parameter passed to function.
- ***FC2_ERROR_INVALID_SETTINGS*** Setting set to camera is invalid.
- ***FC2_ERROR_INVALID_BUS_MANAGER*** Invalid Bus Manager object.
- ***FC2_ERROR_MEMORY_ALLOCATION_FAILED*** Could not allocate memory.
- ***FC2_ERROR_LOW_LEVEL_FAILURE*** Low level error.
- ***FC2_ERROR_NOT_FOUND*** Device not found.
- ***FC2_ERROR_FAILED_GUID*** GUID failure.
- ***FC2_ERROR_INVALID_PACKET_SIZE*** Packet size set to camera is invalid.

FC2_ERROR_INVALID_MODE Invalid mode has been passed to function.

FC2_ERROR_NOT_IN_FORMAT7 Error due to not being in Format7.

FC2_ERROR_NOT_SUPPORTED This feature is unsupported.

FC2_ERROR_TIMEOUT Timeout error.

FC2_ERROR_BUS_MASTER_FAILED Bus Master Failure.

FC2_ERROR_INVALID_GENERATION Generation Count Mismatch.

FC2_ERROR_LUT_FAILED Look Up Table failure.

FC2_ERROR_IIDC_FAILED IIDC failure.

FC2_ERROR_STROBE_FAILED Strobe failure.

FC2_ERROR_TRIGGER_FAILED Trigger failure.

FC2_ERROR_PROPERTY_FAILED Property failure.

FC2_ERROR_PROPERTY_NOT_PRESENT Property is not present.

FC2_ERROR_REGISTER_FAILED Register access failed.

FC2_ERROR_READ_REGISTER_FAILED Register read failed.

FC2_ERROR_WRITE_REGISTER_FAILED Register write failed.

FC2_ERROR_ISOCH_FAILED Isochronous failure.

FC2_ERROR_ISOCH_ALREADY_STARTED Isochronous transfer has already been started.

FC2_ERROR_ISOCH_NOT_STARTED Isochronous transfer has not been started.

FC2_ERROR_ISOCH_START_FAILED Isochronous start failed.

FC2_ERROR_ISOCH_RETRIEVE_BUFFER_FAILED Isochronous retrieve buffer failed.

FC2_ERROR_ISOCH_STOP_FAILED Isochronous stop failed.

FC2_ERROR_ISOCH_SYNC_FAILED Isochronous image synchronization failed.

FC2_ERROR_ISOCH_BANDWIDTH_EXCEEDED Isochronous bandwidth exceeded.

FC2_ERROR_IMAGE_CONVERSION_FAILED Image conversion failed.

FC2_ERROR_IMAGE_LIBRARY_FAILURE Image library failure.

FC2_ERROR_BUFFER_TOO_SMALL Buffer is too small.

FC2_ERROR_IMAGE_CONSISTENCY_ERROR There is an image consistency error.

FC2_ERROR_INCOMPATIBLE_DRIVER The installed driver is not compatible with the library.

FC2_ERROR_FORCE_32BITS

5.2.3.9 enum fc2FrameRate

Enumerator:

FC2_FRAMERATE_1_875	1.875 fps.
FC2_FRAMERATE_3_75	3.75 fps.
FC2_FRAMERATE_7_5	7.5 fps.
FC2_FRAMERATE_15	15 fps.
FC2_FRAMERATE_30	30 fps.
FC2_FRAMERATE_60	60 fps.
FC2_FRAMERATE_120	120 fps.
FC2_FRAMERATE_240	240 fps.
FC2_FRAMERATE_FORMAT7	Custom frame rate for Format7 functionality.
FC2_NUM_FRAMERATES	Number of possible camera frame rates.
FC2_FRAMERATE_FORCE_32BITS	

5.2.3.10 enum fc2GigEPropertyType

Enumerator:

FC2_HEARTBEAT

FC2_HEARTBEAT_TIMEOUT

5.2.3.11 enum fc2GrabMode

Enumerator:

FC2_DROP_FRAMES

FC2_BUFFER_FRAMES

FC2_UNSPECIFIED_GRAB_MODE

FC2_GRAB_MODE_FORCE_32BITS

5.2.3.12 enum fc2GrabTimeout

Enumerator:

FC2_TIMEOUT_NONE

FC2_TIMEOUT_INFINITE

FC2_TIMEOUT_UNSPECIFIED

FC2_GRAB_TIMEOUT_FORCE_32BITS

5.2.3.13 enum fc2ImageFileFormat

Enumerator:

FC2_FROM_FILE_EXT Determine file format from file extension.

FC2_PGM Portable gray map.

FC2_PPM Portable pixmap.

FC2_BMP Bitmap.

FC2_JPEG JPEG.

FC2_JPEG2000 JPEG 2000.

FC2_TIFF Tagged image file format.

FC2_PNG Portable network graphics.

FC2_RAW Raw data.

FC2_IMAGE_FILE_FORMAT_FORCE_32BITS

5.2.3.14 enum fc2InterfaceType

Enumerator:

FC2_INTERFACE_IEEE1394

FC2_INTERFACE_USB_2

FC2_INTERFACE_USB_3

FC2_INTERFACE_GIGE

FC2_INTERFACE_UNKNOWN

FC2_INTERFACE_TYPE_FORCE_32BITS

5.2.3.15 enum fc2Mode

Enumerator:

FC2_MODE_0

FC2_MODE_1

FC2_MODE_2

FC2_MODE_3

FC2_MODE_4

FC2_MODE_5

FC2_MODE_6

FC2_MODE_7

FC2_MODE_8

FC2_MODE_9

FC2_MODE_10

FC2_MODE_11

FC2_MODE_12

FC2_MODE_13

FC2_MODE_14

FC2_MODE_15

FC2_MODE_16

FC2_MODE_17

FC2_MODE_18

FC2_MODE_19

FC2_MODE_20

FC2_MODE_21

FC2_MODE_22

FC2_MODE_23

FC2_MODE_24

FC2_MODE_25

FC2_MODE_26

FC2_MODE_27

FC2_MODE_28

FC2_MODE_29

FC2_MODE_30

FC2_MODE_31

FC2_NUM_MODES Number of modes.

FC2_MODE_FORCE_32BITS

5.2.3.16 enum fc2OSType

Enumerator:

FC2_WINDOWS_X86

FC2_WINDOWS_X64

FC2_LINUX_X86

FC2_LINUX_X64

FC2_MAC

FC2_UNKNOWN_OS

FC2_OSTYPE_FORCE_32BITS

5.2.3.17 enum fc2PCIeBusSpeed

Enumerator:

FC2_PCIE_BUSSPEED_2_5

FC2_PCIE_BUSSPEED_5_0 2.5 Gb/s

FC2_PCIE_BUSSPEED_UNKNOWN 5.0 Gb/s

FC2_PCIE_BUSSPEED_FORCE_32BITS Speed is unknown.

5.2.3.18 enum fc2PixelFormat

Enumerator:

FC2_PIXEL_FORMAT_MONO8 8 bits of mono information.

FC2_PIXEL_FORMAT_411YUV8 YUV 4:1:1.

FC2_PIXEL_FORMAT_422YUV8 YUV 4:2:2.

FC2_PIXEL_FORMAT_444YUV8 YUV 4:4:4.

FC2_PIXEL_FORMAT_RGB8 R = G = B = 8 bits.

FC2_PIXEL_FORMAT_MONO16 16 bits of mono information.

FC2_PIXEL_FORMAT_RGB16 R = G = B = 16 bits.

FC2_PIXEL_FORMAT_S_MONO16 16 bits of signed mono information.

FC2_PIXEL_FORMAT_S_RGB16 R = G = B = 16 bits signed.

FC2_PIXEL_FORMAT_RAW8 8 bit raw data output of sensor.

FC2_PIXEL_FORMAT_RAW16 16 bit raw data output of sensor.

FC2_PIXEL_FORMAT_MONO12 12 bits of mono information.

FC2_PIXEL_FORMAT_RAW12 12 bit raw data output of sensor.

FC2_PIXEL_FORMAT_BGR 24 bit BGR.

FC2_PIXEL_FORMAT_BGRU 32 bit BGRU.

FC2_PIXEL_FORMAT_RGB 24 bit RGB.

FC2_PIXEL_FORMAT_RGBU 32 bit RGBU.

FC2_PIXEL_FORMAT_BGR16 R = G = B = 16 bits.

FC2_PIXEL_FORMAT_BGRU16 64 bit BGRU.

FC2_PIXEL_FORMAT_422YUV8_JPEG JPEG compressed stream.

FC2_NUM_PIXEL_FORMATS Number of pixel formats.

FC2_UNSPECIFIED_PIXEL_FORMAT Unspecified pixel format.

5.2.3.19 enum fc2PropertyType

Enumerator:

- ***FC2_BRIGHTNESS***
- ***FC2_AUTO_EXPOSURE***
- ***FC2_SHARPNESS***
- ***FC2_WHITE_BALANCE***
- ***FC2_HUE***
- ***FC2_SATURATION***
- ***FC2_GAMMA***
- ***FC2_IRIS***
- ***FC2_FOCUS***
- ***FC2_ZOOM***
- ***FC2_PAN***
- ***FC2_TILT***
- ***FC2_SHUTTER***
- ***FC2_GAIN***
- ***FC2_TRIGGER_MODE***
- ***FC2_TRIGGER_DELAY***
- ***FC2_FRAME_RATE***
- ***FC2_TEMPERATURE***
- ***FC2_UNSPECIFIED_PROPERTY_TYPE***
- ***FC2_PROPERTY_TYPE_FORCE_32BITS***

5.2.3.20 enum fc2StatisticsChannel

Enumerator:

- ***FC2_STATISTICS_GREY***
- ***FC2_STATISTICS_RED***
- ***FC2_STATISTICS_GREEN***
- ***FC2_STATISTICS_BLUE***
- ***FC2_STATISTICS_HUE***
- ***FC2_STATISTICS_SATURATION***
- ***FC2_STATISTICS_LIGHTNESS***
- ***FC2_STATISTICS_FORCE_32BITS***

5.2.3.21 enum fc2TIFFCompressionMethod

Enumerator:

- ***FC2_TIFF_NONE***
- ***FC2_TIFF_PACKBITS***
- ***FC2_TIFF_DEFLATE***
- ***FC2_TIFF_ADOBE_DEFLATE***
- ***FC2_TIFF_CCITTFAX3***
- ***FC2_TIFF_CCITTFAX4***
- ***FC2_TIFF_LZW***
- ***FC2_TIFF_JPEG***

5.2.3.22 enum fc2VideoMode

Enumerator:

FC2_VIDEOMODE_160x120YUV444	160x120 YUV444.
FC2_VIDEOMODE_320x240YUV422	320x240 YUV422.
FC2_VIDEOMODE_640x480YUV411	640x480 YUV411.
FC2_VIDEOMODE_640x480YUV422	640x480 YUV422.
FC2_VIDEOMODE_640x480RGB	640x480 24-bit RGB.
FC2_VIDEOMODE_640x480Y8	640x480 8-bit.
FC2_VIDEOMODE_640x480Y16	640x480 16-bit.
FC2_VIDEOMODE_800x600YUV422	800x600 YUV422.
FC2_VIDEOMODE_800x600RGB	800x600 RGB.
FC2_VIDEOMODE_800x600Y8	800x600 8-bit.
FC2_VIDEOMODE_800x600Y16	800x600 16-bit.
FC2_VIDEOMODE_1024x768YUV422	1024x768 YUV422.
FC2_VIDEOMODE_1024x768RGB	1024x768 RGB.
FC2_VIDEOMODE_1024x768Y8	1024x768 8-bit.
FC2_VIDEOMODE_1024x768Y16	1024x768 16-bit.
FC2_VIDEOMODE_1280x960YUV422	1280x960 YUV422.
FC2_VIDEOMODE_1280x960RGB	1280x960 RGB.
FC2_VIDEOMODE_1280x960Y8	1280x960 8-bit.
FC2_VIDEOMODE_1280x960Y16	1280x960 16-bit.
FC2_VIDEOMODE_1600x1200YUV422	1600x1200 YUV422.
FC2_VIDEOMODE_1600x1200RGB	1600x1200 RGB.
FC2_VIDEOMODE_1600x1200Y8	1600x1200 8-bit.
FC2_VIDEOMODE_1600x1200Y16	1600x1200 16-bit.
FC2_VIDEOMODE_FORMAT7	Custom video mode for Format7 functionality.
FC2_NUM_VIDEOMODES	Number of possible video modes.
FC2_VIDEOMODE_FORCE_32BITS	

5.3 FlyCapture2GUI_C.h File Reference

Functions

- FLYCAPTURE2_C_API fc2Error fc2CreateGUIContext (fc2GuiContext ∗pContext)

  Create a GUI context.
- FLYCAPTURE2_C_API fc2Error fc2DestroyGUIContext (fc2GuiContext context)

  Destroy a GUI context.
- FLYCAPTURE2_C_API void fc2GUIConnect (fc2GuiContext context, fc2Context cameraContext)

  Connect GUI context to a camera context.
- FLYCAPTURE2_C_API void fc2GUIDisconnect (fc2GuiContext context)

  Disconnect GUI context from camera.
- FLYCAPTURE2_C_API void fc2Disonnect (fc2GuiContext context) __attribute__((deprecated))

  Disconnect GUI context from camera.
- FLYCAPTURE2_C_API void fc2Show (fc2GuiContext context)

  Show the GUI.
- FLYCAPTURE2_C_API void fc2Hide (fc2GuiContext context)

  Hide the GUI.
- FLYCAPTURE2_C_API BOOL fc2IsVisible (fc2GuiContext context)

  Check if the GUI is visible.
- FLYCAPTURE2_C_API void fc2ShowModal (fc2GuiContext context, BOOL ∗pOkSelected, fc2PGRGuid ∗guidArray, unsigned int ∗size)

  Show the camera selection dialog.

5.3.1 Function Documentation

5.3.1.1 FLYCAPTURE2_C_API fc2Error fc2CreateGUIContext (fc2GuiContext ∗ *pContext*)

Create a GUI context.

Parameters

pContext	Pointer to context to be created.

Returns

An Error indicating the success or failure of the function.

5.3.1.2 FLYCAPTURE2_C_API fc2Error fc2DestroyGUIContext (fc2GuiContext *context*)

Destroy a GUI context.

Parameters

context	Context to be destroyed.

Returns

An Error indicating the success or failure of the function.

5.3.1.3 FLYCAPTURE2_C_API void fc2Disonnect (fc2GuiContext *context*)

Disconnect GUI context from camera.

Parameters

context	GUI context to disconnect.

Returns

An Error indicating the success or failure of the function.

Deprecated This method is deprecated and will be removed in a future FlyCapture2 release. Please use fc2GUIDisconnect instead.

5.3.1.4 FLYCAPTURE2_C_API void fc2GUIConnect (fc2GuiContext *context,* fc2Context *cameraContext*)

Connect GUI context to a camera context.

Parameters

context	GUI context to connect.
camera-Context	Camera context to connect.

Returns

An Error indicating the success or failure of the function.

5.3.1.5 FLYCAPTURE2_C_API void fc2GUIDisconnect (fc2GuiContext *context*)

Disconnect GUI context from camera.

Parameters

context	GUI context to disconnect.

Returns

An Error indicating the success or failure of the function.

5.3.1.6 FLYCAPTURE2_C_API void fc2Hide (**fc2GuiContext** *context*)

Hide the GUI.

Parameters

context	Pointer to context to hide.

Returns

An Error indicating the success or failure of the function.

5.3.1.7 FLYCAPTURE2_C_API **BOOL** fc2IsVisible (**fc2GuiContext** *context*)

Check if the GUI is visible.

Parameters

context	Pointer to context to show.

Returns

Whether the GUI is visible.

5.3.1.8 FLYCAPTURE2_C_API void fc2Show (**fc2GuiContext** *context*)

Show the GUI.

Parameters

context	Pointer to context to show.

Returns

An Error indicating the success or failure of the function.

5.3.1.9 FLYCAPTURE2_C_API void fc2ShowModal (**fc2GuiContext** *context,* **BOOL** ∗ *pOkSelected,* **fc2PGRGuid** ∗ *guidArray,* unsigned int ∗ *size*)

Show the camera selection dialog.

Parameters

context	Pointer to context to show.
pOkSelected	Whether Ok (true) or Cancel (false) was clicked.
guidArray	Array of PGRGuids containing the selected cameras.
size	Size of PGRGuid array.

5.4 FlyCapture2Internal_C.h File Reference

Data Structures

- struct fc2InternalContext
- struct fc2InternalGuiContext
- struct fc2InternalImageCallback

Functions

- bool IsContextValid (fc2Context context)
- bool IsGuiContextValid (fc2GuiContext context)
- void SyncCppImageToStruct (fc2Image *pImage)

5.4.1 Function Documentation

5.4.1.1 bool IsContextValid (**fc2Context** *context*) `[inline]`

5.4.1.2 bool IsGuiContextValid (**fc2GuiContext** *context*) `[inline]`

5.4.1.3 void SyncCppImageToStruct (**fc2Image** ∗ *pImage*) `[inline]`

5.5 FlyCapture2Platform_C.h File Reference

Defines

- #define FLYCAPTURE2_C_API
- #define FLYCAPTURE2_C_CALL_CONVEN

5.5.1 Define Documentation

5.5.1.1 #define FLYCAPTURE2_C_API

5.5.1.2 #define FLYCAPTURE2_C_CALL_CONVEN

5.6 MultiSyncLibrary_C.h File Reference

Functions

- MULTISYNCLIBRARY_C_API syncError syncCreateContext (syncContext *pContext)

  Create a Sync context for MultiSync Library.
- MULTISYNCLIBRARY_C_API syncError syncDestroyContext (syncContext context)

  Destory the sync context.
- MULTISYNCLIBRARY_C_API syncError syncStart (syncContext context)

  Start the sync progress.
- MULTISYNCLIBRARY_C_API syncError syncStop (syncContext context)

  Stop the sync progress.
- MULTISYNCLIBRARY_C_API syncError syncRescanMasterTimingBus (syncContext context)

  Scan newly connected or removed timing bus (for corss-PC syncing only)
- MULTISYNCLIBRARY_C_API syncMessage syncGetStatus (syncContext context)

  Start the sync progress.
- MULTISYNCLIBRARY_C_API double syncGetTimeSinceSynced (syncContext context)

  Time since sync started.
- MULTISYNCLIBRARY_C_API bool syncIsTimingBusConnected (syncContext context)

  Whether syncing across PCs.
- MULTISYNCLIBRARY_C_API bool syncEnableCrossPCSynchronization (syncContext context)

  Enable across pc synchronization support.
- MULTISYNCLIBRARY_C_API bool syncDisableCrossPCSynchronization (syncContext context)

  Disable across pc synchronization support.
- MULTISYNCLIBRARY_C_API bool syncQueryCrossPCSynchronizationSetting (syncContext context)

  Query cross pc synchronizaion support status.

5.6.1 Function Documentation

5.6.1.1 MULTISYNCLIBRARY_C_API syncError syncCreateContext (syncContext * *pContext*)

Create a Sync context for MultiSync Library.

This call must be made before any other calls that use a context will succeed.

Parameters

pContext	A pointer to the syncContext to be created.

Returns

A syncError indicating the success or failure of the function.

5.6.1.2 MULTISYNCLIBRARY_C_API syncError syncDestroyContext (syncContext *context*)

Destory the sync context.

This must be called when the user is finished with the context in order to prevent memory leaks.

Parameters

context	The syncContext to be destoryed.

Returns

A syncError indicating the success or failure of the function.

5.6.1.3 MULTISYNCLIBRARY_C_API bool syncDisableCrossPCSynchronization (syncContext *context*)

Disable across pc synchronization support.

Parameters

context	The syncContext to be used.

Returns

True if operation was successful

5.6.1.4 MULTISYNCLIBRARY_C_API bool syncEnableCrossPCSynchronization (syncContext *context*)

Enable across pc synchronization support.

Parameters

context	The syncContext to be used.

Returns

True if operation was successful

5.6.1.5 MULTISYNCLIBRARY_C_API **syncMessage** syncGetStatus (**syncContext** *context*)

Start the sync progress.

Parameters

context	The syncContext to be used.

Returns

A syncMessage indicating the sync status.

5.6.1.6 MULTISYNCLIBRARY_C_API double syncGetTimeSinceSynced (**syncContext** *context*)

Time since sync started.

Parameters

context	The syncContext to be used.

Returns

Time sinced synced.

5.6.1.7 MULTISYNCLIBRARY_C_API bool syncIsTimingBusConnected (**syncContext** *context*)

Whether syncing across PCs.

Parameters

context	The syncContext to be used.

Returns

True if its syncing across PC

5.6.1.8 MULTISYNCLIBRARY_C_API bool syncQueryCrossPCSynchronizationSetting (**syncContext** *context*)

Query cross pc synchronizaion support status.

Parameters

context	The syncContext to be used.

Returns

True if cross pc synchronization was supported

5.6.1.9 MULTISYNCLIBRARY_C_API syncError syncRescanMasterTimingBus (syncContext *context*)

Scan newly connected or removed timing bus (for corss-PC syncing only)

Parameters

context	The syncContext to be used.

Returns

A syncError indicating the success or failure of the function.

5.6.1.10 MULTISYNCLIBRARY_C_API syncError syncStart (syncContext *context*)

Start the sync progress.

Parameters

context	The syncContext to be used.

Returns

A syncError indicating the success or failure of the function.

5.6.1.11 MULTISYNCLIBRARY_C_API syncError syncStop (syncContext *context*)

Stop the sync progress.

Parameters

context	The syncContext to be used.

Returns

A syncError indicating the success or failure of the function.

5.7 MultiSyncLibraryDefs_C.h File Reference

Defines

- #define FALSE 0
- #define TRUE 1
- #define FULL_32BIT_VALUE 0x7FFFFFFF
- #define MAX_STRING_LENGTH 512

Typedefs

- typedef int BOOL
- typedef void ∗ syncContext

  A context to the MultiSync C library.

Enumerations

- enum syncError { SYNC_ERROR_OK = 0, SYNC_ERROR_FAILED, SYN-C_ERROR_ALREADY_STARTED, SYNC_ERROR_ALREADY_STOPPED, S-YNC_ERROR_CONTEXT_NOT_INITIALIZED, SYNC_ERROR_UNKNOWN_E-RROR }
- enum syncMessage { SYNC_MESSAGE_OK = 0, SYNC_MESSAGE_FAILED, SYNC_MESSAGE_STARTED, SYNC_MESSAGE_STOPPED, SYNC_MESS-AGE_SYNCING, SYNC_MESSAGE_NOMASTER, SYNC_MESSAGE_THRE-AD_ERROR, SYNC_MESSAGE_DEVICE_ERROR, SYNC_MESSAGE_NOT-_ENOUGH_DEVICES, SYNC_MESSAGE_BUS_RESET, SYNC_MESSAGE_-NOT_INITIALIZED, SYNC_MESSAGE_UNKNOWN_ERROR }

5.7.1 Define Documentation

5.7.1.1 #define FALSE 0

5.7.1.2 #define FULL_32BIT_VALUE 0x7FFFFFFF

5.7.1.3 #define MAX_STRING_LENGTH 512

5.7.1.4 #define TRUE 1

5.7.2 Typedef Documentation

5.7.2.1 typedef int BOOL

5.7.2.2 typedef void∗ syncContext

A context to the MultiSync C library.

It must be created before performing any calls to the library.

5.7.3 Enumeration Type Documentation

5.7.3.1 enum syncError

Enumerator:

- ***SYNC_ERROR_OK***
- ***SYNC_ERROR_FAILED***
- ***SYNC_ERROR_ALREADY_STARTED***
- ***SYNC_ERROR_ALREADY_STOPPED***
- ***SYNC_ERROR_CONTEXT_NOT_INITIALIZED***
- ***SYNC_ERROR_UNKNOWN_ERROR***

5.7.3.2 enum syncMessage

Enumerator:

- ***SYNC_MESSAGE_OK***
- ***SYNC_MESSAGE_FAILED***
- ***SYNC_MESSAGE_STARTED***
- ***SYNC_MESSAGE_STOPPED***
- ***SYNC_MESSAGE_SYNCING***
- ***SYNC_MESSAGE_NOMASTER***
- ***SYNC_MESSAGE_THREAD_ERROR***
- ***SYNC_MESSAGE_DEVICE_ERROR***
- ***SYNC_MESSAGE_NOT_ENOUGH_DEVICES***
- ***SYNC_MESSAGE_BUS_RESET***
- ***SYNC_MESSAGE_NOT_INITIALIZED***
- ***SYNC_MESSAGE_UNKNOWN_ERROR***

5.8 MultiSyncLibraryPlatform_C.h File Reference

Defines

- #define MULTISYNCLIBRARY_C_API
- #define MULTISYNCLIBRARY_C_CALL_CONVEN

5.8.1 Define Documentation

5.8.1.1 #define MULTISYNCLIBRARY_C_API

5.8.1.2 #define MULTISYNCLIBRARY_C_CALL_CONVEN

Index

FC2_ARRIVAL
FlyCapture2Defs_C.h, 98
FC2_AUTO_EXPOSURE
FlyCapture2Defs_C.h, 106
FC2_BANDWIDTH_ALLOCATION_FORCE_32BITS
FlyCapture2Defs_C.h, 98
FC2_BANDWIDTH_ALLOCATION_OFF
FlyCapture2Defs_C.h, 98
FC2_BANDWIDTH_ALLOCATION_ON
FlyCapture2Defs_C.h, 98
FC2_BANDWIDTH_ALLOCATION_UNSPECIFIED
FlyCapture2Defs_C.h, 98
FC2_BANDWIDTH_ALLOCATION_UNSUPPORTED
FlyCapture2Defs_C.h, 98
FC2_BMP
FlyCapture2Defs_C.h, 103
FC2_BRIGHTNESS
FlyCapture2Defs_C.h, 106
FC2_BT_BGGR
FlyCapture2Defs_C.h, 98
FC2_BT_FORCE_32BITS
FlyCapture2Defs_C.h, 98
FC2_BT_GBRG
FlyCapture2Defs_C.h, 98
FC2_BT_GRBG
FlyCapture2Defs_C.h, 98
FC2_BT_NONE
FlyCapture2Defs_C.h, 98
FC2_BT_RGGB
FlyCapture2Defs_C.h, 98
FC2_BUFFER_FRAMES
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_10000BASE_T
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_1000BASE_T
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_100BASE_T
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_10BASE_T
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_ANY
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_FORCE_32BITS
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S100
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S1600
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S200
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S3200
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S400
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S480
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S5000
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S800
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_SPEED_UNKNOWN
FlyCapture2Defs_C.h, 99
FC2_BUSSPEED_S_FASTEST
FlyCapture2Defs_C.h, 99
FC2_BUS_RESET
FlyCapture2Defs_C.h, 98
FC2_BYTE_ORDER_BIG_ENDIAN
FlyCapture2Defs_C.h, 99
FC2_BYTE_ORDER_FORCE_32BITS
FlyCapture2Defs_C.h, 99
FC2_BYTE_ORDER_LITTLE_ENDIAN
FlyCapture2Defs_C.h, 99
FC2_CALLBACK_TYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 98
FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS
FlyCapture2Defs_C.h, 99
FC2_DEFAULT

FlyCapture2Defs_C.h, 99
FC2_DIRECTIONAL
FlyCapture2Defs_C.h, 99
FC2_DRIVER_1394_CAM
FlyCapture2Defs_C.h, 100
FC2_DRIVER_1394_JUJU
FlyCapture2Defs_C.h, 100
FC2_DRIVER_1394_PRO
FlyCapture2Defs_C.h, 100
FC2_DRIVER_1394_RAW1394
FlyCapture2Defs_C.h, 100
FC2_DRIVER_1394_VIDEO1394
FlyCapture2Defs_C.h, 100
FC2_DRIVER_FORCE_32BITS
FlyCapture2Defs_C.h, 100
FC2_DRIVER_GIGE_FILTER
FlyCapture2Defs_C.h, 100
FC2_DRIVER_GIGE_NONE
FlyCapture2Defs_C.h, 100
FC2_DRIVER_GIGE_PRO
FlyCapture2Defs_C.h, 100
FC2_DRIVER_UNKNOWN
FlyCapture2Defs_C.h, 100
FC2_DRIVER_USB3_PRO
FlyCapture2Defs_C.h, 100
FC2_DRIVER_USB_CAM
FlyCapture2Defs_C.h, 100
FC2_DRIVER_USB_NONE
FlyCapture2Defs_C.h, 100
FC2_DROP_FRAMES
FlyCapture2Defs_C.h, 102
FC2_EDGE_SENSING
FlyCapture2Defs_C.h, 99
FC2_ERROR_BUFFER_TOO_SMALL
FlyCapture2Defs_C.h, 101
FC2_ERROR_BUS_MASTER_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_FAILED
FlyCapture2Defs_C.h, 100
FC2_ERROR_FAILED_BUS_MASTER_-
CONNECTION
FlyCapture2Defs_C.h, 100
FC2_ERROR_FAILED_GUID
FlyCapture2Defs_C.h, 100
FC2_ERROR_FORCE_32BITS
FlyCapture2Defs_C.h, 101
FC2_ERROR_IIDC_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_IMAGE_CONSISTENCY_-
ERROR
FlyCapture2Defs_C.h, 101
FC2_ERROR_IMAGE_CONVERSION_-
FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_IMAGE_LIBRARY_FAILU-
RE
FlyCapture2Defs_C.h, 101
FC2_ERROR_INCOMPATIBLE_DRIVER
FlyCapture2Defs_C.h, 101
FC2_ERROR_INIT_FAILED
FlyCapture2Defs_C.h, 100
FC2_ERROR_INVALID_BUS_MANAGE-
R
FlyCapture2Defs_C.h, 100
FC2_ERROR_INVALID_GENERATION
FlyCapture2Defs_C.h, 101
FC2_ERROR_INVALID_MODE
FlyCapture2Defs_C.h, 100
FC2_ERROR_INVALID_PACKET_SIZE
FlyCapture2Defs_C.h, 100
FC2_ERROR_INVALID_PARAMETER
FlyCapture2Defs_C.h, 100
FC2_ERROR_INVALID_SETTINGS
FlyCapture2Defs_C.h, 100
FC2_ERROR_ISOCH_ALREADY_STAR-
TED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_BANDWIDTH_E-
XCEEDED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_NOT_STARTED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_RETRIEVE_BUF-
FER_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_START_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_STOP_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_ISOCH_SYNC_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_LOW_LEVEL_FAILURE
FlyCapture2Defs_C.h, 100
FC2_ERROR_LUT_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_MEMORY_ALLOCATION-
_FAILED
FlyCapture2Defs_C.h, 100

FC2_ERROR_NOT_CONNECTED
FlyCapture2Defs_C.h, 100
FC2_ERROR_NOT_FOUND
FlyCapture2Defs_C.h, 100
FC2_ERROR_NOT_IMPLEMENTED
FlyCapture2Defs_C.h, 100
FC2_ERROR_NOT_INTITIALIZED
FlyCapture2Defs_C.h, 100
FC2_ERROR_NOT_IN_FORMAT7
FlyCapture2Defs_C.h, 101
FC2_ERROR_NOT_SUPPORTED
FlyCapture2Defs_C.h, 101
FC2_ERROR_OK
FlyCapture2Defs_C.h, 100
FC2_ERROR_PROPERTY_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_PROPERTY_NOT_PRES-ENT
FlyCapture2Defs_C.h, 101
FC2_ERROR_READ_REGISTER_FAIL-ED
FlyCapture2Defs_C.h, 101
FC2_ERROR_REGISTER_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_STROBE_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_TIMEOUT
FlyCapture2Defs_C.h, 101
FC2_ERROR_TRIGGER_FAILED
FlyCapture2Defs_C.h, 101
FC2_ERROR_UNDEFINED
FlyCapture2Defs_C.h, 100
FC2_ERROR_WRITE_REGISTER_FAIL-ED
FlyCapture2Defs_C.h, 101
FC2_FOCUS
FlyCapture2Defs_C.h, 106
FC2_FRAMERATE_120
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_15
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_1_875
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_240
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_30
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_3_75
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_60
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_7_5
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_FRAMERATE_FORMAT7
FlyCapture2Defs_C.h, 102
FC2_FRAME_RATE
FlyCapture2Defs_C.h, 106
FC2_FROM_FILE_EXT
FlyCapture2Defs_C.h, 103
FC2_GAIN
FlyCapture2Defs_C.h, 106
FC2_GAMMA
FlyCapture2Defs_C.h, 106
FC2_GRAB_MODE_FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_GRAB_TIMEOUT_FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_HEARTBEAT
FlyCapture2Defs_C.h, 102
FC2_HEARTBEAT_TIMEOUT
FlyCapture2Defs_C.h, 102
FC2_HQ_LINEAR
FlyCapture2Defs_C.h, 99
FC2_HUE
FlyCapture2Defs_C.h, 106
FC2_IMAGE_FILE_FORMAT_FORCE_-32BITS
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_GIGE
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_IEEE1394
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_TYPE_FORCE_32BI-TS
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_UNKNOWN
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_USB_2
FlyCapture2Defs_C.h, 103
FC2_INTERFACE_USB_3
FlyCapture2Defs_C.h, 103
FC2_IPP
FlyCapture2Defs_C.h, 99
FC2_IRIS
FlyCapture2Defs_C.h, 106
FC2_JPEG
FlyCapture2Defs_C.h, 103
FC2_JPEG2000

FlyCapture2Defs_C.h, 103
FC2_LINUX_X64
FlyCapture2Defs_C.h, 104
FC2_LINUX_X86
FlyCapture2Defs_C.h, 104
FC2_MAC
FlyCapture2Defs_C.h, 104
FC2_MODE_0
FlyCapture2Defs_C.h, 103
FC2_MODE_1
FlyCapture2Defs_C.h, 103
FC2_MODE_10
FlyCapture2Defs_C.h, 103
FC2_MODE_11
FlyCapture2Defs_C.h, 103
FC2_MODE_12
FlyCapture2Defs_C.h, 104
FC2_MODE_13
FlyCapture2Defs_C.h, 104
FC2_MODE_14
FlyCapture2Defs_C.h, 104
FC2_MODE_15
FlyCapture2Defs_C.h, 104
FC2_MODE_16
FlyCapture2Defs_C.h, 104
FC2_MODE_17
FlyCapture2Defs_C.h, 104
FC2_MODE_18
FlyCapture2Defs_C.h, 104
FC2_MODE_19
FlyCapture2Defs_C.h, 104
FC2_MODE_2
FlyCapture2Defs_C.h, 103
FC2_MODE_20
FlyCapture2Defs_C.h, 104
FC2_MODE_21
FlyCapture2Defs_C.h, 104
FC2_MODE_22
FlyCapture2Defs_C.h, 104
FC2_MODE_23
FlyCapture2Defs_C.h, 104
FC2_MODE_24
FlyCapture2Defs_C.h, 104
FC2_MODE_25
FlyCapture2Defs_C.h, 104
FC2_MODE_26
FlyCapture2Defs_C.h, 104
FC2_MODE_27
FlyCapture2Defs_C.h, 104
FC2_MODE_28
FlyCapture2Defs_C.h, 104
FC2_MODE_29
FlyCapture2Defs_C.h, 104
FC2_MODE_3
FlyCapture2Defs_C.h, 103
FC2_MODE_30
FlyCapture2Defs_C.h, 104
FC2_MODE_31
FlyCapture2Defs_C.h, 104
FC2_MODE_4
FlyCapture2Defs_C.h, 103
FC2_MODE_5
FlyCapture2Defs_C.h, 103
FC2_MODE_6
FlyCapture2Defs_C.h, 103
FC2_MODE_7
FlyCapture2Defs_C.h, 103
FC2_MODE_8
FlyCapture2Defs_C.h, 103
FC2_MODE_9
FlyCapture2Defs_C.h, 103
FC2_MODE_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_NEAREST_NEIGHBOR_FAST
FlyCapture2Defs_C.h, 99
FC2_NO_COLOR_PROCESSING
FlyCapture2Defs_C.h, 99
FC2_NUM_FRAMERATES
FlyCapture2Defs_C.h, 102
FC2_NUM_MODES
FlyCapture2Defs_C.h, 104
FC2_NUM_PIXEL_FORMATS
FlyCapture2Defs_C.h, 105
FC2_NUM_VIDEOMODES
FlyCapture2Defs_C.h, 107
FC2_OSTYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_PAN
FlyCapture2Defs_C.h, 106
FC2_PCIE_BUSSPEED_2_5
FlyCapture2Defs_C.h, 105
FC2_PCIE_BUSSPEED_5_0
FlyCapture2Defs_C.h, 105
FC2_PCIE_BUSSPEED_FORCE_32BIT-
S
FlyCapture2Defs_C.h, 105
FC2_PCIE_BUSSPEED_UNKNOWN
FlyCapture2Defs_C.h, 105
FC2_PGM
FlyCapture2Defs_C.h, 103

FC2_PIXEL_FORMAT_411YUV8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_422YUV8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_422YUV8_JPEG
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_444YUV8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_BGR
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_BGR16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_BGRU
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_BGRU16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_MONO12
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_MONO16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_MONO8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RAW12
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RAW16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RAW8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RGB
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RGB16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RGB8
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_RGBU
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_S_MONO16
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_S_RGB16
FlyCapture2Defs_C.h, 105
FC2_PNG
FlyCapture2Defs_C.h, 103
FC2_PPM
FlyCapture2Defs_C.h, 103
FC2_PROPERTY_TYPE_FORCE_32BI-TS
FlyCapture2Defs_C.h, 106
FC2_RAW
FlyCapture2Defs_C.h, 103
FC2_REMOVAL
FlyCapture2Defs_C.h, 98
FC2_RIGOROUS
FlyCapture2Defs_C.h, 99
FC2_SATURATION
FlyCapture2Defs_C.h, 106
FC2_SHARPNESS
FlyCapture2Defs_C.h, 106
FC2_SHUTTER
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_BLUE
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_FORCE_32BITS
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_GREEN
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_GREY
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_HUE
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_LIGHTNESS
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_RED
FlyCapture2Defs_C.h, 106
FC2_STATISTICS_SATURATION
FlyCapture2Defs_C.h, 106
FC2_TEMPERATURE
FlyCapture2Defs_C.h, 106
FC2_TIFF
FlyCapture2Defs_C.h, 103
FC2_TIFF_ADOBE_DEFLATE
FlyCapture2Defs_C.h, 107
FC2_TIFF_CCITTFAX3
FlyCapture2Defs_C.h, 107
FC2_TIFF_CCITTFAX4
FlyCapture2Defs_C.h, 107
FC2_TIFF_DEFLATE
FlyCapture2Defs_C.h, 107
FC2_TIFF_JPEG
FlyCapture2Defs_C.h, 107
FC2_TIFF_LZW
FlyCapture2Defs_C.h, 107
FC2_TIFF_NONE
FlyCapture2Defs_C.h, 107
FC2_TIFF_PACKBITS
FlyCapture2Defs_C.h, 107
FC2_TILT
FlyCapture2Defs_C.h, 106
FC2_TIMEOUT_INFINITE
FlyCapture2Defs_C.h, 102
FC2_TIMEOUT_NONE

FlyCapture2Defs_C.h, 102
FC2_TIMEOUT_UNSPECIFIED
FlyCapture2Defs_C.h, 102
FC2_TRIGGER_DELAY
FlyCapture2Defs_C.h, 106
FC2_TRIGGER_MODE
FlyCapture2Defs_C.h, 106
FC2_UNKNOWN_OS
FlyCapture2Defs_C.h, 104
FC2_UNSPECIFIED_GRAB_MODE
FlyCapture2Defs_C.h, 102
FC2_UNSPECIFIED_PIXEL_FORMAT
FlyCapture2Defs_C.h, 105
FC2_UNSPECIFIED_PROPERTY_TYPE
FlyCapture2Defs_C.h, 106
FC2_VIDEOMODE_1024x768RGB
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1024x768Y16
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1024x768Y8
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1024x768YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1280x960RGB
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1280x960Y16
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1280x960Y8
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1280x960YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1600x1200RGB
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1600x1200Y16
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1600x1200Y8
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1600x1200YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_160x120YUV444
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_320x240YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_640x480RGB
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_640x480Y16
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_640x480Y8
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_640x480YUV411
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_640x480YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_800x600RGB
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_800x600Y16
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_800x600Y8
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_800x600YUV422
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_FORCE_32BITS
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_FORMAT7
FlyCapture2Defs_C.h, 107
FC2_WHITE_BALANCE
FlyCapture2Defs_C.h, 106
FC2_WINDOWS_X64
FlyCapture2Defs_C.h, 104
FC2_WINDOWS_X86
FlyCapture2Defs_C.h, 104
FC2_ZOOM
FlyCapture2Defs_C.h, 106
FlyCapture2Defs_C.h
FC2_ARRIVAL, 98
FC2_AUTO_EXPOSURE, 106
FC2_BANDWIDTH_ALLOCATION_-FORCE_32BITS, 98
FC2_BANDWIDTH_ALLOCATION_-OFF, 98
FC2_BANDWIDTH_ALLOCATION_-ON, 98
FC2_BANDWIDTH_ALLOCATION_-UNSPECIFIED, 98
FC2_BANDWIDTH_ALLOCATION_-UNSUPPORTED, 98
FC2_BMP, 103
FC2_BRIGHTNESS, 106
FC2_BT_BGGR, 98
FC2_BT_FORCE_32BITS, 98
FC2_BT_GBRG, 98
FC2_BT_GRBG, 98
FC2_BT_NONE, 98
FC2_BT_RGGB, 98
FC2_BUFFER_FRAMES, 102
FC2_BUSSPEED_10000BASE_T, 99
FC2_BUSSPEED_1000BASE_T, 99
FC2_BUSSPEED_100BASE_T, 99
FC2_BUSSPEED_10BASE_T, 99

FC2_BUSSPEED_ANY, 99
FC2_BUSSPEED_FORCE_32BITS, 99
FC2_BUSSPEED_S100, 99
FC2_BUSSPEED_S1600, 99
FC2_BUSSPEED_S200, 99
FC2_BUSSPEED_S3200, 99
FC2_BUSSPEED_S400, 99
FC2_BUSSPEED_S480, 99
FC2_BUSSPEED_S5000, 99
FC2_BUSSPEED_S800, 99
FC2_BUSSPEED_SPEED_UNKNOWN, 99
FC2_BUSSPEED_S_FASTEST, 99
FC2_BUS_RESET, 98
FC2_BYTE_ORDER_BIG_ENDIAN, 99
FC2_BYTE_ORDER_FORCE_32BITS, 99
FC2_BYTE_ORDER_LITTLE_ENDIAN, 99
FC2_CALLBACK_TYPE_FORCE_32BITS, 98
FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS, 99
FC2_DEFAULT, 99
FC2_DIRECTIONAL, 99
FC2_DRIVER_1394_CAM, 100
FC2_DRIVER_1394_JUJU, 100
FC2_DRIVER_1394_PRO, 100
FC2_DRIVER_1394_RAW1394, 100
FC2_DRIVER_1394_VIDEO1394, 100
FC2_DRIVER_FORCE_32BITS, 100
FC2_DRIVER_GIGE_FILTER, 100
FC2_DRIVER_GIGE_NONE, 100
FC2_DRIVER_GIGE_PRO, 100
FC2_DRIVER_UNKNOWN, 100
FC2_DRIVER_USB3_PRO, 100
FC2_DRIVER_USB_CAM, 100
FC2_DRIVER_USB_NONE, 100
FC2_DROP_FRAMES, 102
FC2_EDGE_SENSING, 99
FC2_ERROR_BUFFER_TOO_SMALL, 101
FC2_ERROR_BUS_MASTER_FAILED, 101
FC2_ERROR_FAILED, 100
FC2_ERROR_FAILED_BUS_MASTER_CONNECTION, 100
FC2_ERROR_FAILED_GUID, 100
FC2_ERROR_FORCE_32BITS, 101
FC2_ERROR_IIDC_FAILED, 101
FC2_ERROR_IMAGE_CONSISTENCY_ERROR, 101
FC2_ERROR_IMAGE_CONVERSION_FAILED, 101
FC2_ERROR_IMAGE_LIBRARY_FAILURE, 101
FC2_ERROR_INCOMPATIBLE_DRIVER, 101
FC2_ERROR_INIT_FAILED, 100
FC2_ERROR_INVALID_BUS_MANAGER, 100
FC2_ERROR_INVALID_GENERATION, 101
FC2_ERROR_INVALID_MODE, 100
FC2_ERROR_INVALID_PACKET_SIZE, 100
FC2_ERROR_INVALID_PARAMETER, 100
FC2_ERROR_INVALID_SETTINGS, 100
FC2_ERROR_ISOCH_ALREADY_STARTED, 101
FC2_ERROR_ISOCH_BANDWIDTH_EXCEEDED, 101
FC2_ERROR_ISOCH_FAILED, 101
FC2_ERROR_ISOCH_NOT_STARTED, 101
FC2_ERROR_ISOCH_RETRIEVE_BUFFER_FAILED, 101
FC2_ERROR_ISOCH_START_FAILED, 101
FC2_ERROR_ISOCH_STOP_FAILED, 101
FC2_ERROR_ISOCH_SYNC_FAILED, 101
FC2_ERROR_LOW_LEVEL_FAILURE, 100
FC2_ERROR_LUT_FAILED, 101
FC2_ERROR_MEMORY_ALLOCATION_FAILED, 100
FC2_ERROR_NOT_CONNECTED, 100
FC2_ERROR_NOT_FOUND, 100
FC2_ERROR_NOT_IMPLEMENTED, 100

FC2_ERROR_NOT_INTITIALIZED, 100
FC2_ERROR_NOT_IN_FORMAT7, 101
FC2_ERROR_NOT_SUPPORTED, 101
FC2_ERROR_OK, 100
FC2_ERROR_PROPERTY_FAILED, 101
FC2_ERROR_PROPERTY_NOT_PRESENT, 101
FC2_ERROR_READ_REGISTER_FAILED, 101
FC2_ERROR_REGISTER_FAILED, 101
FC2_ERROR_STROBE_FAILED, 101
FC2_ERROR_TIMEOUT, 101
FC2_ERROR_TRIGGER_FAILED, 101
FC2_ERROR_UNDEFINED, 100
FC2_ERROR_WRITE_REGISTER_FAILED, 101
FC2_FOCUS, 106
FC2_FRAMERATE_120, 102
FC2_FRAMERATE_15, 102
FC2_FRAMERATE_1_875, 102
FC2_FRAMERATE_240, 102
FC2_FRAMERATE_30, 102
FC2_FRAMERATE_3_75, 102
FC2_FRAMERATE_60, 102
FC2_FRAMERATE_7_5, 102
FC2_FRAMERATE_FORCE_32BITS, 102
FC2_FRAMERATE_FORMAT7, 102
FC2_FRAME_RATE, 106
FC2_FROM_FILE_EXT, 103
FC2_GAIN, 106
FC2_GAMMA, 106
FC2_GRAB_MODE_FORCE_32BITS, 102
FC2_GRAB_TIMEOUT_FORCE_32BITS, 102
FC2_HEARTBEAT, 102
FC2_HEARTBEAT_TIMEOUT, 102
FC2_HQ_LINEAR, 99
FC2_HUE, 106
FC2_IMAGE_FILE_FORMAT_FORCE_32BITS, 103
FC2_INTERFACE_GIGE, 103
FC2_INTERFACE_IEEE1394, 103
FC2_INTERFACE_TYPE_FORCE_32BITS, 103
FC2_INTERFACE_UNKNOWN, 103
FC2_INTERFACE_USB_2, 103
FC2_INTERFACE_USB_3, 103
FC2_IPP, 99
FC2_IRIS, 106
FC2_JPEG, 103
FC2_JPEG2000, 103
FC2_LINUX_X64, 104
FC2_LINUX_X86, 104
FC2_MAC, 104
FC2_MODE_0, 103
FC2_MODE_1, 103
FC2_MODE_10, 103
FC2_MODE_11, 103
FC2_MODE_12, 104
FC2_MODE_13, 104
FC2_MODE_14, 104
FC2_MODE_15, 104
FC2_MODE_16, 104
FC2_MODE_17, 104
FC2_MODE_18, 104
FC2_MODE_19, 104
FC2_MODE_2, 103
FC2_MODE_20, 104
FC2_MODE_21, 104
FC2_MODE_22, 104
FC2_MODE_23, 104
FC2_MODE_24, 104
FC2_MODE_25, 104
FC2_MODE_26, 104
FC2_MODE_27, 104
FC2_MODE_28, 104
FC2_MODE_29, 104
FC2_MODE_3, 103
FC2_MODE_30, 104
FC2_MODE_31, 104
FC2_MODE_4, 103
FC2_MODE_5, 103
FC2_MODE_6, 103
FC2_MODE_7, 103
FC2_MODE_8, 103
FC2_MODE_9, 103
FC2_MODE_FORCE_32BITS, 104
FC2_NEAREST_NEIGHBOR_FAST, 99
FC2_NO_COLOR_PROCESSING, 99

FC2_NUM_FRAMERATES, 102
FC2_NUM_MODES, 104
FC2_NUM_PIXEL_FORMATS, 105
FC2_NUM_VIDEOMODES, 107
FC2_OSTYPE_FORCE_32BITS, 104
FC2_PAN, 106
FC2_PCIE_BUSSPEED_2_5, 105
FC2_PCIE_BUSSPEED_5_0, 105
FC2_PCIE_BUSSPEED_FORCE_32BITS, 105
FC2_PCIE_BUSSPEED_UNKNOWN, 105
FC2_PGM, 103
FC2_PIXEL_FORMAT_411YUV8, 105
FC2_PIXEL_FORMAT_422YUV8, 105
FC2_PIXEL_FORMAT_422YUV8_JPEG, 105
FC2_PIXEL_FORMAT_444YUV8, 105
FC2_PIXEL_FORMAT_BGR, 105
FC2_PIXEL_FORMAT_BGR16, 105
FC2_PIXEL_FORMAT_BGRU, 105
FC2_PIXEL_FORMAT_BGRU16, 105
FC2_PIXEL_FORMAT_MONO12, 105
FC2_PIXEL_FORMAT_MONO16, 105
FC2_PIXEL_FORMAT_MONO8, 105
FC2_PIXEL_FORMAT_RAW12, 105
FC2_PIXEL_FORMAT_RAW16, 105
FC2_PIXEL_FORMAT_RAW8, 105
FC2_PIXEL_FORMAT_RGB, 105
FC2_PIXEL_FORMAT_RGB16, 105
FC2_PIXEL_FORMAT_RGB8, 105
FC2_PIXEL_FORMAT_RGBU, 105
FC2_PIXEL_FORMAT_S_MONO16, 105
FC2_PIXEL_FORMAT_S_RGB16, 105
FC2_PNG, 103
FC2_PPM, 103
FC2_PROPERTY_TYPE_FORCE_32BITS, 106
FC2_RAW, 103
FC2_REMOVAL, 98
FC2_RIGOROUS, 99
FC2_SATURATION, 106
FC2_SHARPNESS, 106
FC2_SHUTTER, 106
FC2_STATISTICS_BLUE, 106
FC2_STATISTICS_FORCE_32BITS, 106
FC2_STATISTICS_GREEN, 106
FC2_STATISTICS_GREY, 106
FC2_STATISTICS_HUE, 106
FC2_STATISTICS_LIGHTNESS, 106
FC2_STATISTICS_RED, 106
FC2_STATISTICS_SATURATION, 106
FC2_TEMPERATURE, 106
FC2_TIFF, 103
FC2_TIFF_ADOBE_DEFLATE, 107
FC2_TIFF_CCITTFAX3, 107
FC2_TIFF_CCITTFAX4, 107
FC2_TIFF_DEFLATE, 107
FC2_TIFF_JPEG, 107
FC2_TIFF_LZW, 107
FC2_TIFF_NONE, 107
FC2_TIFF_PACKBITS, 107
FC2_TILT, 106
FC2_TIMEOUT_INFINITE, 102
FC2_TIMEOUT_NONE, 102
FC2_TIMEOUT_UNSPECIFIED, 102
FC2_TRIGGER_DELAY, 106
FC2_TRIGGER_MODE, 106
FC2_UNKNOWN_OS, 104
FC2_UNSPECIFIED_GRAB_MODE, 102
FC2_UNSPECIFIED_PIXEL_FORMAT, 105
FC2_UNSPECIFIED_PROPERTY_TYPE, 106
FC2_VIDEOMODE_1024x768RGB, 107
FC2_VIDEOMODE_1024x768Y16, 107
FC2_VIDEOMODE_1024x768Y8, 107
FC2_VIDEOMODE_1024x768YUV422, 107
FC2_VIDEOMODE_1280x960RGB, 107
FC2_VIDEOMODE_1280x960Y16, 107

FC2_VIDEOMODE_1280x960Y8, 107
FC2_VIDEOMODE_1280x960YUV422, 107
FC2_VIDEOMODE_1600x1200RGB, 107
FC2_VIDEOMODE_1600x1200Y16, 107
FC2_VIDEOMODE_1600x1200Y8, 107
FC2_VIDEOMODE_1600x1200YUV422, 107
FC2_VIDEOMODE_160x120YUV444, 107
FC2_VIDEOMODE_320x240YUV422, 107
FC2_VIDEOMODE_640x480RGB, 107
FC2_VIDEOMODE_640x480Y16, 107
FC2_VIDEOMODE_640x480Y8, 107
FC2_VIDEOMODE_640x480YUV411, 107
FC2_VIDEOMODE_640x480YUV422, 107
FC2_VIDEOMODE_800x600RGB, 107
FC2_VIDEOMODE_800x600Y16, 107
FC2_VIDEOMODE_800x600Y8, 107
FC2_VIDEOMODE_800x600YUV422, 107
FC2_VIDEOMODE_FORCE_32BITS, 107
FC2_VIDEOMODE_FORMAT7, 107
FC2_WHITE_BALANCE, 106
FC2_WINDOWS_X64, 104
FC2_WINDOWS_X86, 104
FC2_ZOOM, 106
MultiSyncLibraryDefs_C.h
SYNC_ERROR_ALREADY_STARTED, 117
SYNC_ERROR_ALREADY_STOPPED, 117
SYNC_ERROR_CONTEXT_NOT_INITIALIZED, 117
SYNC_ERROR_FAILED, 117
SYNC_ERROR_OK, 117
SYNC_ERROR_UNKNOWN_ERROR, 117
SYNC_MESSAGE_BUS_RESET, 117
SYNC_MESSAGE_DEVICE_ERROR, 117
SYNC_MESSAGE_FAILED, 117
SYNC_MESSAGE_NOMASTER, 117
SYNC_MESSAGE_NOT_ENOUGH_DEVICES, 117
SYNC_MESSAGE_NOT_INITIALIZED, 117
SYNC_MESSAGE_OK, 117
SYNC_MESSAGE_STARTED, 117
SYNC_MESSAGE_STOPPED, 117
SYNC_MESSAGE_SYNCING, 117
SYNC_MESSAGE_THREAD_ERROR, 117
SYNC_MESSAGE_UNKNOWN_ERROR, 117
SYNC_ERROR_ALREADY_STARTED
MultiSyncLibraryDefs_C.h, 117
SYNC_ERROR_ALREADY_STOPPED
MultiSyncLibraryDefs_C.h, 117
SYNC_ERROR_CONTEXT_NOT_INITIALIZED
MultiSyncLibraryDefs_C.h, 117
SYNC_ERROR_FAILED
MultiSyncLibraryDefs_C.h, 117
SYNC_ERROR_OK
MultiSyncLibraryDefs_C.h, 117
SYNC_ERROR_UNKNOWN_ERROR
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_BUS_RESET
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_DEVICE_ERROR
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_FAILED
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_NOMASTER
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_NOT_ENOUGH_DEVICES
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_NOT_INITIALIZED
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_OK
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_STARTED
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_STOPPED

MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_SYNCING
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_THREAD_ERROR
MultiSyncLibraryDefs_C.h, 117
SYNC_MESSAGE_UNKNOWN_ERROR
MultiSyncLibraryDefs_C.h, 117
BOOL
FlyCapture2Defs_C.h, 97
MultiSyncLibraryDefs_C.h, 116
FALSE
FlyCapture2Defs_C.h, 97
MultiSyncLibraryDefs_C.h, 116
FULL_32BIT_VALUE
FlyCapture2Defs_C.h, 97
MultiSyncLibraryDefs_C.h, 116
FlyCapture2Defs_C.h, 92
BOOL, 97
FALSE, 97
TRUE, 97
fc2AVIContext, 97
fc2AsyncCommandCallback, 97
fc2BandwidthAllocation, 98
fc2BayerTileFormat, 98
fc2BusCallbackType, 98
fc2BusEventCallback, 97
fc2BusSpeed, 98
fc2ByteOrder, 99
fc2CallbackHandle, 97
fc2ColorProcessingAlgorithm, 99
fc2Context, 97
fc2DriverType, 99
fc2Error, 100
fc2FrameRate, 101
fc2GigEPropertyType, 102
fc2GrabMode, 102
fc2GrabTimeout, 102
fc2GuiContext, 97
fc2ImageEventCallback, 97
fc2ImageFileFormat, 102
fc2ImageImpl, 98
fc2ImageStatisticsContext, 98
fc2InterfaceType, 103
fc2Mode, 103
fc2OSType, 104
fc2PCIeBusSpeed, 104
fc2PixelFormat, 105
fc2PropertyType, 105
fc2StatisticsChannel, 106
fc2TIFFCompressionMethod, 106
fc2VideoMode, 107
FlyCapture2GUI_C.h, 108
fc2CreateGUIContext, 108
fc2DestroyGUIContext, 108
fc2Disonnect, 109
fc2GUIConnect, 109
fc2GUIDisconnect, 109
fc2Hide, 110
fc2IsVisible, 110
fc2Show, 110
fc2ShowModal, 110
FlyCapture2Internal_C.h, 111
IsContextValid, 111
IsGuiContextValid, 111
SyncCppImageToStruct, 111
FlyCapture2Platform_C.h, 111
FlyCapture2_C.h, 39
fc2AVIAppend, 48
fc2AVIClose, 48
fc2AVIOpen, 49
fc2CalculateImageStatistics, 49
fc2Connect, 49
fc2ConvertImage, 50
fc2ConvertImageTo, 50
fc2CreateAVI, 50
fc2CreateContext, 51
fc2CreateGigEContext, 51
fc2CreateImage, 51
fc2CreateImageStatistics, 52
fc2DestroyAVI, 52
fc2DestroyContext, 52
fc2DestroyImage, 53
fc2DestroyImageStatistics, 53
fc2DetermineBitsPerPixel, 53
fc2Disconnect, 54
fc2DiscoverGigECameras, 54
fc2EnableLUT, 54
fc2ErrorToDescription, 55
fc2FireBusReset, 55
fc2FireSoftwareTrigger, 55
fc2FireSoftwareTriggerBroadcast, 56
fc2ForceAllIPAddressesAutomatically, 56
fc2ForceIPAddressAutomatically, 56
fc2ForceIPAddressToCamera, 56
fc2GetActiveLUTBank, 57
fc2GetCameraFromIPAddress, 58
fc2GetCameraFromIndex, 57
fc2GetCameraFromSerialNumber, 58

fc2GetCameraInfo, 58
fc2GetCameraSerialNumberFromIndex, 59
fc2GetChannelStatus, 59
fc2GetConfiguration, 59
fc2GetCycleTime, 60
fc2GetDefaultColorProcessing, 60
fc2GetDefaultOutputFormat, 60
fc2GetDeviceFromIndex, 61
fc2GetEmbeddedImageInfo, 61
fc2GetFormat7Configuration, 61
fc2GetFormat7Info, 62
fc2GetGPIOPinDirection, 63
fc2GetGigEConfig, 62
fc2GetGigEImageBinningSettings, 62
fc2GetGigEImageSettings, 62
fc2GetGigEImageSettingsInfo, 62
fc2GetGigEImagingMode, 62
fc2GetGigEProperty, 62
fc2GetGigEStreamChannelInfo, 63
fc2GetImageData, 63
fc2GetImageStatistics, 64
fc2GetImageTimeStamp, 64
fc2GetInterfaceTypeFromGuid, 64
fc2GetLUTBankInfo, 65
fc2GetLUTChannel, 66
fc2GetLUTInfo, 66
fc2GetLibraryVersion, 65
fc2GetMemoryChannel, 66
fc2GetMemoryChannelInfo, 67
fc2GetNumOfCameras, 67
fc2GetNumOfDevices, 67
fc2GetNumStreamChannels, 68
fc2GetProperty, 68
fc2GetPropertyInfo, 68
fc2GetRegisterString, 68
fc2GetStrobe, 69
fc2GetStrobeInfo, 69
fc2GetSystemInfo, 69
fc2GetTriggerDelay, 70
fc2GetTriggerDelayInfo, 70
fc2GetTriggerMode, 70
fc2GetTriggerModeInfo, 70
fc2GetVideoModeAndFrameRate, 71
fc2GetVideoModeAndFrameRateInfo, 71
fc2H264Open, 72
fc2IsCameraControlable, 72
fc2LaunchBrowser, 72
fc2LaunchCommand, 73
fc2LaunchCommandAsync, 73
fc2LaunchHelp, 73
fc2MJPGOpen, 73
fc2QueryGigEImagingMode, 74
fc2ReadGVCPMemory, 74
fc2ReadGVCPRegister, 74
fc2ReadGVCPRegisterBlock, 75
fc2ReadRegister, 75
fc2ReadRegisterBlock, 75
fc2RegisterCallback, 76
fc2RescanBus, 76
fc2RestoreFromMemoryChannel, 76
fc2RetrieveBuffer, 77
fc2SaveImage, 77
fc2SaveImageWithOption, 77
fc2SaveToMemoryChannel, 78
fc2SetActiveLUTBank, 78
fc2SetCallback, 78
fc2SetChannelStatus, 79
fc2SetConfiguration, 79
fc2SetDefaultColorProcessing, 79
fc2SetDefaultOutputFormat, 80
fc2SetEmbeddedImageInfo, 80
fc2SetFormat7Configuration, 81
fc2SetFormat7ConfigurationPacket, 81
fc2SetGPIOPinDirection, 82
fc2SetGPIOPinDirectionBroadcast, 82
fc2SetGigEConfig, 81
fc2SetGigEImageBinningSettings, 81
fc2SetGigEImageSettings, 81
fc2SetGigEImagingMode, 81
fc2SetGigEProperty, 82
fc2SetGigEStreamChannelInfo, 82
fc2SetImageData, 83
fc2SetImageDimensions, 83
fc2SetLUTChannel, 83
fc2SetProperty, 84
fc2SetPropertyBroadcast, 84
fc2SetStrobe, 85
fc2SetStrobeBroadcast, 85
fc2SetTriggerDelay, 85
fc2SetTriggerDelayBroadcast, 86
fc2SetTriggerMode, 86
fc2SetTriggerModeBroadcast, 86
fc2SetUserBuffers, 86
fc2SetVideoModeAndFrameRate, 87

fc2StartCapture, 87
fc2StartCaptureCallback, 88
fc2StartSyncCapture, 88
fc2StartSyncCaptureCallback, 88
fc2StopCapture, 89
fc2UnregisterCallback, 89
fc2ValidateFormat7Settings, 89
fc2WriteGVCPMemory, 90
fc2WriteGVCPRegister, 90
fc2WriteGVCPRegisterBlock, 91
fc2WriteGVCPRegisterBroadcast, 91
fc2WriteRegister, 91
fc2WriteRegisterBlock, 92
fc2WriteRegisterBroadcast, 92
GPIOPinState
fc2EmbeddedImageInfo, 14
IsContextValid
FlyCapture2Internal_C.h, 111
IsGuiContextValid
FlyCapture2Internal_C.h, 111
MultiSyncLibraryDefs_C.h, 115
BOOL, 116
FALSE, 116
TRUE, 116
syncContext, 116
syncError, 117
syncMessage, 117
MultiSyncLibraryPlatform_C.h, 117
MultiSyncLibrary_C.h, 111
syncCreateContext, 112
syncDestroyContext, 113
syncDisableCrossPCSynchronization, 113
syncEnableCrossPCSynchronization, 113
syncGetStatus, 113
syncGetTimeSinceSynced, 114
syncIsTimingBusConnected, 114
syncQueryCrossPCSynchronization-Setting, 114
syncRescanMasterTimingBus, 115
syncStart, 115
syncStop, 115
ROIPosition
fc2EmbeddedImageInfo, 14
SyncCppImageToStruct
FlyCapture2Internal_C.h, 111
TRUE
FlyCapture2Defs_C.h, 97
MultiSyncLibraryDefs_C.h, 116
absControl
fc2TriggerDelay, 34
absMax
fc2TriggerDelayInfo, 35
absMin
fc2TriggerDelayInfo, 35
absValSupported
fc2TriggerDelayInfo, 35
absValue
fc2TriggerDelay, 34
applicationIPAddress
fc2CameraInfo, 9
applicationPort
fc2CameraInfo, 9
asyncBusSpeed
fc2Config, 11
autoManualMode
fc2TriggerDelay, 34
autoSupported
fc2TriggerDelayInfo, 35
available
fc2EmbeddedImageInfoProperty, 14

bandwidthAllocation
fc2Config, 11
bayerFormat
fc2Image, 22
bayerTileFormat
fc2CameraInfo, 9
binaryFile
fc2PGMOption, 28
fc2PPMOption, 30
bitrate
fc2H264Option, 21
brightness
fc2EmbeddedImageInfo, 13
build
fc2Version, 37
busNumber
fc2CameraInfo, 9
byteOrder
fc2SystemInfo, 32

ccpStatus
fc2CameraInfo, 9
chipIdHi
fc2ConfigROM, 12
chipIdLo
fc2ConfigROM, 12
cols

fc2Image, 22
compression
fc2TIFFOption, 33
compressionLevel
fc2PNGOption, 29
configROM
fc2CameraInfo, 9
cpuDescription
fc2SystemInfo, 32
cycleCount
fc2TimeStamp, 33
cycleOffset
fc2TimeStamp, 33
cycleSeconds
fc2TimeStamp, 33

dataSize
fc2Image, 22
defaultGateway
fc2CameraInfo, 9
delay
fc2StrobeControl, 30
destinationIpAddress
fc2GigEStreamChannel, 20
doNotFragment
fc2GigEStreamChannel, 20
driverList
fc2SystemInfo, 32
driverName
fc2CameraInfo, 9
driverType
fc2CameraInfo, 9
duration
fc2StrobeControl, 30

embeddedBrightness
fc2ImageMetadata, 23
embeddedExposure
fc2ImageMetadata, 23
embeddedFrameCounter
fc2ImageMetadata, 23
embeddedGPIOPinState
fc2ImageMetadata, 23
embeddedGain
fc2ImageMetadata, 23
embeddedROIPosition
fc2ImageMetadata, 23
embeddedShutter
fc2ImageMetadata, 23
embeddedStrobePattern
fc2ImageMetadata, 23
embeddedTimeStamp
fc2ImageMetadata, 23
embeddedWhiteBalance
fc2ImageMetadata, 23
enablePacketResend
fc2GigEConfig, 17
enabled
fc2LUTData, 27
exposure
fc2EmbeddedImageInfo, 13

fc2AVIAppend
FlyCapture2_C.h, 48
fc2AVIClose
FlyCapture2_C.h, 48
fc2AVIContext
FlyCapture2Defs_C.h, 97
fc2AVIOpen
FlyCapture2_C.h, 49
fc2AVIOption, 7
frameRate, 7
reserved, 7
fc2AsyncCommandCallback
FlyCapture2Defs_C.h, 97
fc2BandwidthAllocation
FlyCapture2Defs_C.h, 98
fc2BayerTileFormat
FlyCapture2Defs_C.h, 98
fc2BusCallbackType
FlyCapture2Defs_C.h, 98
fc2BusEventCallback
FlyCapture2Defs_C.h, 97
fc2BusSpeed
FlyCapture2Defs_C.h, 98
fc2ByteOrder
FlyCapture2Defs_C.h, 99
fc2CalculateImageStatistics
FlyCapture2_C.h, 49
fc2CallbackHandle
FlyCapture2Defs_C.h, 97
fc2CameraInfo, 8
applicationIPAddress, 9
applicationPort, 9
bayerTileFormat, 9
busNumber, 9
ccpStatus, 9
configROM, 9
defaultGateway, 9
driverName, 9

driverType, 9
firmwareBuildTime, 9
firmwareVersion, 9
gigEMajorVersion, 9
gigEMinorVersion, 9
iidcVer, 10
interfaceType, 10
ipAddress, 10
isColorCamera, 10
macAddress, 10
maximumBusSpeed, 10
modelName, 10
nodeNumber, 10
pcieBusSpeed, 10
reserved, 10
sensorInfo, 10
sensorResolution, 10
serialNumber, 10
subnetMask, 10
userDefinedName, 10
vendorName, 10
xmlURL1, 10
xmlURL2, 10
fc2ColorProcessingAlgorithm
FlyCapture2Defs_C.h, 99
fc2Config, 10
asyncBusSpeed, 11
bandwidthAllocation, 11
grabMode, 11
grabTimeout, 11
isochBusSpeed, 11
minNumImageNotifications, 11
numBuffers, 11
numImageNotifications, 11
registerTimeout, 11
registerTimeoutRetries, 11
reserved, 11
fc2ConfigROM, 11
chipIdHi, 12
chipIdLo, 12
nodeVendorId, 12
pszKeyword, 12
reserved, 12
unitSWVer, 12
unitSpecId, 12
unitSubSWVer, 12
vendorUniqueInfo_0, 12
vendorUniqueInfo_1, 12
vendorUniqueInfo_2, 12
vendorUniqueInfo_3, 12
fc2Connect
FlyCapture2_C.h, 49
fc2Context
FlyCapture2Defs_C.h, 97
fc2ConvertImage
FlyCapture2_C.h, 50
fc2ConvertImageTo
FlyCapture2_C.h, 50
fc2CreateAVI
FlyCapture2_C.h, 50
fc2CreateContext
FlyCapture2_C.h, 51
fc2CreateGUIContext
FlyCapture2GUI_C.h, 108
fc2CreateGigEContext
FlyCapture2_C.h, 51
fc2CreateImage
FlyCapture2_C.h, 51
fc2CreateImageStatistics
FlyCapture2_C.h, 52
fc2DestroyAVI
FlyCapture2_C.h, 52
fc2DestroyContext
FlyCapture2_C.h, 52
fc2DestroyGUIContext
FlyCapture2GUI_C.h, 108
fc2DestroyImage
FlyCapture2_C.h, 53
fc2DestroyImageStatistics
FlyCapture2_C.h, 53
fc2DetermineBitsPerPixel
FlyCapture2_C.h, 53
fc2Disconnect
FlyCapture2_C.h, 54
fc2DiscoverGigECameras
FlyCapture2_C.h, 54
fc2Disonnect
FlyCapture2GUI_C.h, 109
fc2DriverType
FlyCapture2Defs_C.h, 99
fc2EmbeddedImageInfo, 13
GPIOPinState, 14
ROIPosition, 14
brightness, 13
exposure, 13
frameCounter, 14
gain, 14
shutter, 14
strobePattern, 14
timestamp, 14

whiteBalance, 14
fc2EmbeddedImageInfoProperty, 14
available, 14
onOff, 14
fc2EnableLUT
FlyCapture2_C.h, 54
fc2Error
FlyCapture2Defs_C.h, 100
fc2ErrorToDescription
FlyCapture2_C.h, 55
fc2FireBusReset
FlyCapture2_C.h, 55
fc2FireSoftwareTrigger
FlyCapture2_C.h, 55
fc2FireSoftwareTriggerBroadcast
FlyCapture2_C.h, 56
fc2ForceAllIPAddressesAutomatically
FlyCapture2_C.h, 56
fc2ForceIPAddressAutomatically
FlyCapture2_C.h, 56
fc2ForceIPAddressToCamera
FlyCapture2_C.h, 56
fc2Format7ImageSettings, 14
height, 15
mode, 15
offsetX, 15
offsetY, 15
pixelFormat, 15
reserved, 15
width, 15
fc2Format7Info, 15
imageHStepSize, 16
imageVStepSize, 16
maxHeight, 16
maxPacketSize, 16
maxWidth, 16
minPacketSize, 16
mode, 16
offsetHStepSize, 16
offsetVStepSize, 16
packetSize, 16
percentage, 16
pixelFormatBitField, 16
reserved, 16
vendorPixelFormatBitField, 16
fc2Format7PacketInfo, 16
maxBytesPerPacket, 16
recommendedBytesPerPacket, 17
reserved, 17
unitBytesPerPacket, 17
fc2FrameRate
FlyCapture2Defs_C.h, 101
fc2GUIConnect
FlyCapture2GUI_C.h, 109
fc2GUIDisconnect
FlyCapture2GUI_C.h, 109
fc2GetActiveLUTBank
FlyCapture2_C.h, 57
fc2GetCameraFromIPAddress
FlyCapture2_C.h, 58
fc2GetCameraFromIndex
FlyCapture2_C.h, 57
fc2GetCameraFromSerialNumber
FlyCapture2_C.h, 58
fc2GetCameraInfo
FlyCapture2_C.h, 58
fc2GetCameraSerialNumberFromIndex
FlyCapture2_C.h, 59
fc2GetChannelStatus
FlyCapture2_C.h, 59
fc2GetConfiguration
FlyCapture2_C.h, 59
fc2GetCycleTime
FlyCapture2_C.h, 60
fc2GetDefaultColorProcessing
FlyCapture2_C.h, 60
fc2GetDefaultOutputFormat
FlyCapture2_C.h, 60
fc2GetDeviceFromIndex
FlyCapture2_C.h, 61
fc2GetEmbeddedImageInfo
FlyCapture2_C.h, 61
fc2GetFormat7Configuration
FlyCapture2_C.h, 61
fc2GetFormat7Info
FlyCapture2_C.h, 62
fc2GetGPIOPinDirection
FlyCapture2_C.h, 63
fc2GetGigEConfig
FlyCapture2_C.h, 62
fc2GetGigEImageBinningSettings
FlyCapture2_C.h, 62
fc2GetGigEImageSettings
FlyCapture2_C.h, 62
fc2GetGigEImageSettingsInfo
FlyCapture2_C.h, 62
fc2GetGigEImagingMode
FlyCapture2_C.h, 62
fc2GetGigEProperty
FlyCapture2_C.h, 62

fc2GetGigEStreamChannelInfo
FlyCapture2_C.h, 63
fc2GetImageData
FlyCapture2_C.h, 63
fc2GetImageStatistics
FlyCapture2_C.h, 64
fc2GetImageTimeStamp
FlyCapture2_C.h, 64
fc2GetInterfaceTypeFromGuid
FlyCapture2_C.h, 64
fc2GetLUTBankInfo
FlyCapture2_C.h, 65
fc2GetLUTChannel
FlyCapture2_C.h, 66
fc2GetLUTInfo
FlyCapture2_C.h, 66
fc2GetLibraryVersion
FlyCapture2_C.h, 65
fc2GetMemoryChannel
FlyCapture2_C.h, 66
fc2GetMemoryChannelInfo
FlyCapture2_C.h, 67
fc2GetNumOfCameras
FlyCapture2_C.h, 67
fc2GetNumOfDevices
FlyCapture2_C.h, 67
fc2GetNumStreamChannels
FlyCapture2_C.h, 68
fc2GetProperty
FlyCapture2_C.h, 68
fc2GetPropertyInfo
FlyCapture2_C.h, 68
fc2GetRegisterString
FlyCapture2_C.h, 68
fc2GetStrobe
FlyCapture2_C.h, 69
fc2GetStrobeInfo
FlyCapture2_C.h, 69
fc2GetSystemInfo
FlyCapture2_C.h, 69
fc2GetTriggerDelay
FlyCapture2_C.h, 70
fc2GetTriggerDelayInfo
FlyCapture2_C.h, 70
fc2GetTriggerMode
FlyCapture2_C.h, 70
fc2GetTriggerModeInfo
FlyCapture2_C.h, 70
fc2GetVideoModeAndFrameRate
FlyCapture2_C.h, 71
fc2GetVideoModeAndFrameRateInfo
FlyCapture2_C.h, 71
fc2GigEConfig, 17
enablePacketResend, 17
maxPacketsToResend, 17
reserved, 17
timeoutForPacketResend, 17
fc2GigEImageSettings, 18
height, 18
offsetX, 18
offsetY, 18
pixelFormat, 18
reserved, 18
width, 18
fc2GigEImageSettingsInfo, 18
imageHStepSize, 19
imageVStepSize, 19
maxHeight, 19
maxWidth, 19
offsetHStepSize, 19
offsetVStepSize, 19
pixelFormatBitField, 19
reserved, 19
vendorPixelFormatBitField, 19
fc2GigEProperty, 19
isReadable, 19
isWritable, 19
max, 19
min, 19
propType, 19
reserved, 20
value, 20
fc2GigEPropertyType
FlyCapture2Defs_C.h, 102
fc2GigEStreamChannel, 20
destinationIpAddress, 20
doNotFragment, 20
hostPost, 21
interPacketDelay, 21
networkInterfaceIndex, 21
packetSize, 21
reserved, 21
sourcePort, 21
fc2GrabMode
FlyCapture2Defs_C.h, 102
fc2GrabTimeout
FlyCapture2Defs_C.h, 102
fc2GuiContext
FlyCapture2Defs_C.h, 97
fc2H264Open

FlyCapture2_C.h, 72
fc2H264Option, 21
bitrate, 21
frameRate, 21
height, 21
reserved, 21
width, 21
fc2Hide
FlyCapture2GUI_C.h, 110
fc2IPAddress, 25
octets, 25
fc2Image, 22
bayerFormat, 22
cols, 22
dataSize, 22
format, 22
imageImpl, 22
pData, 22
receivedDataSize, 22
rows, 22
stride, 22
fc2ImageEventCallback
FlyCapture2Defs_C.h, 97
fc2ImageFileFormat
FlyCapture2Defs_C.h, 102
fc2ImageImpl
FlyCapture2Defs_C.h, 98
fc2ImageMetadata, 22
embeddedBrightness, 23
embeddedExposure, 23
embeddedFrameCounter, 23
embeddedGPIOPinState, 23
embeddedGain, 23
embeddedROIPosition, 23
embeddedShutter, 23
embeddedStrobePattern, 23
embeddedTimeStamp, 23
embeddedWhiteBalance, 23
reserved, 23
fc2ImageStatisticsContext
FlyCapture2Defs_C.h, 98
fc2InterfaceType
FlyCapture2Defs_C.h, 103
fc2InternalContext, 23
pBusMgr, 24
pCamera, 24
fc2InternalGuiContext, 24
pCameraControlDlg, 24
pCameraSelectionDlg, 24
fc2InternalImageCallback, 25
pCallback, 25
pCallbackData, 25
fc2IsCameraControlable
FlyCapture2_C.h, 72
fc2IsVisible
FlyCapture2GUI_C.h, 110
fc2JPEGOption, 26
progressive, 26
quality, 26
reserved, 26
fc2JPG2Option, 26
quality, 26
reserved, 26
fc2LUTData, 27
enabled, 27
inputBitDepth, 27
numBanks, 27
numChannels, 27
numEntries, 27
outputBitDepth, 27
reserved, 27
supported, 27
fc2LaunchBrowser
FlyCapture2_C.h, 72
fc2LaunchCommand
FlyCapture2_C.h, 73
fc2LaunchCommandAsync
FlyCapture2_C.h, 73
fc2LaunchHelp
FlyCapture2_C.h, 73
fc2MACAddress, 27
octets, 27
fc2MJPGOpen
FlyCapture2_C.h, 73
fc2MJPGOption, 28
frameRate, 28
quality, 28
reserved, 28
fc2Mode
FlyCapture2Defs_C.h, 103
fc2OSType
FlyCapture2Defs_C.h, 104
fc2PCIeBusSpeed
FlyCapture2Defs_C.h, 104
fc2PGMOption, 28
binaryFile, 28
reserved, 28
fc2PGRGuid, 29
value, 29
fc2PNGOption, 29

compressionLevel, 29
interlaced, 29
reserved, 29
fc2PPMOption, 30
binaryFile, 30
reserved, 30
fc2PixelFormat
FlyCapture2Defs_C.h, 105
fc2PropertyType
FlyCapture2Defs_C.h, 105
fc2QueryGigEImagingMode
FlyCapture2_C.h, 74
fc2ReadGVCPMemory
FlyCapture2_C.h, 74
fc2ReadGVCPRegister
FlyCapture2_C.h, 74
fc2ReadGVCPRegisterBlock
FlyCapture2_C.h, 75
fc2ReadRegister
FlyCapture2_C.h, 75
fc2ReadRegisterBlock
FlyCapture2_C.h, 75
fc2RegisterCallback
FlyCapture2_C.h, 76
fc2RescanBus
FlyCapture2_C.h, 76
fc2RestoreFromMemoryChannel
FlyCapture2_C.h, 76
fc2RetrieveBuffer
FlyCapture2_C.h, 77
fc2SaveImage
FlyCapture2_C.h, 77
fc2SaveImageWithOption
FlyCapture2_C.h, 77
fc2SaveToMemoryChannel
FlyCapture2_C.h, 78
fc2SetActiveLUTBank
FlyCapture2_C.h, 78
fc2SetCallback
FlyCapture2_C.h, 78
fc2SetChannelStatus
FlyCapture2_C.h, 79
fc2SetConfiguration
FlyCapture2_C.h, 79
fc2SetDefaultColorProcessing
FlyCapture2_C.h, 79
fc2SetDefaultOutputFormat
FlyCapture2_C.h, 80
fc2SetEmbeddedImageInfo
FlyCapture2_C.h, 80
fc2SetFormat7Configuration
FlyCapture2_C.h, 81
fc2SetFormat7ConfigurationPacket
FlyCapture2_C.h, 81
fc2SetGPIOPinDirection
FlyCapture2_C.h, 82
fc2SetGPIOPinDirectionBroadcast
FlyCapture2_C.h, 82
fc2SetGigEConfig
FlyCapture2_C.h, 81
fc2SetGigEImageBinningSettings
FlyCapture2_C.h, 81
fc2SetGigEImageSettings
FlyCapture2_C.h, 81
fc2SetGigEImagingMode
FlyCapture2_C.h, 81
fc2SetGigEProperty
FlyCapture2_C.h, 82
fc2SetGigEStreamChannelInfo
FlyCapture2_C.h, 82
fc2SetImageData
FlyCapture2_C.h, 83
fc2SetImageDimensions
FlyCapture2_C.h, 83
fc2SetLUTChannel
FlyCapture2_C.h, 83
fc2SetProperty
FlyCapture2_C.h, 84
fc2SetPropertyBroadcast
FlyCapture2_C.h, 84
fc2SetStrobe
FlyCapture2_C.h, 85
fc2SetStrobeBroadcast
FlyCapture2_C.h, 85
fc2SetTriggerDelay
FlyCapture2_C.h, 85
fc2SetTriggerDelayBroadcast
FlyCapture2_C.h, 86
fc2SetTriggerMode
FlyCapture2_C.h, 86
fc2SetTriggerModeBroadcast
FlyCapture2_C.h, 86
fc2SetUserBuffers
FlyCapture2_C.h, 86
fc2SetVideoModeAndFrameRate
FlyCapture2_C.h, 87
fc2Show
FlyCapture2GUI_C.h, 110
fc2ShowModal
FlyCapture2GUI_C.h, 110

fc2StartCapture
    FlyCapture2_C.h, 87
fc2StartCaptureCallback
    FlyCapture2_C.h, 88
fc2StartSyncCapture
    FlyCapture2_C.h, 88
fc2StartSyncCaptureCallback
    FlyCapture2_C.h, 88
fc2StatisticsChannel
    FlyCapture2Defs_C.h, 106
fc2StopCapture
    FlyCapture2_C.h, 89
fc2StrobeControl, 30
    delay, 30
    duration, 30
    onOff, 30
    polarity, 30
    reserved, 30
    source, 30
fc2StrobeInfo, 31
    maxValue, 31
    minValue, 31
    onOffSupported, 31
    polaritySupported, 31
    present, 31
    readOutSupported, 31
    reserved, 31
    source, 31
fc2SystemInfo, 31
    byteOrder, 32
    cpuDescription, 32
    driverList, 32
    gpuDescription, 32
    libraryList, 32
    numCpuCores, 32
    osDescription, 32
    osType, 32
    reserved, 32
    screenHeight, 32
    screenWidth, 32
    sysMemSize, 32
fc2TIFFCompressionMethod
    FlyCapture2Defs_C.h, 106
fc2TIFFOption, 32
    compression, 33
    reserved, 33
fc2TimeStamp, 33
    cycleCount, 33
    cycleOffset, 33
    cycleSeconds, 33
    microSeconds, 33
    reserved, 33
    seconds, 33
fc2TriggerDelay, 33
    absControl, 34
    absValue, 34
    autoManualMode, 34
    onOff, 34
    onePush, 34
    present, 34
    reserved, 34
    type, 34
    valueA, 34
    valueB, 34
fc2TriggerDelayInfo, 34
    absMax, 35
    absMin, 35
    absValSupported, 35
    autoSupported, 35
    manualSupported, 35
    max, 35
    min, 35
    onOffSupported, 35
    onePushSupported, 35
    pUnitAbbr, 35
    pUnits, 35
    present, 35
    readOutSupported, 35
    reserved, 35
    type, 35
fc2TriggerMode, 36
    mode, 36
    onOff, 36
    parameter, 36
    polarity, 36
    reserved, 36
    source, 36
fc2TriggerModeInfo, 36
    modeMask, 37
    onOffSupported, 37
    polaritySupported, 37
    present, 37
    readOutSupported, 37
    reserved, 37
    softwareTriggerSupported, 37
    sourceMask, 37
    valueReadable, 37
fc2UnregisterCallback
    FlyCapture2_C.h, 89
fc2ValidateFormat7Settings

FlyCapture2_C.h, 89
fc2Version, 37
build, 37
major, 37
minor, 37
type, 37
fc2VideoMode
FlyCapture2Defs_C.h, 107
fc2WriteGVCPMemory
FlyCapture2_C.h, 90
fc2WriteGVCPRegister
FlyCapture2_C.h, 90
fc2WriteGVCPRegisterBlock
FlyCapture2_C.h, 91
fc2WriteGVCPRegisterBroadcast
FlyCapture2_C.h, 91
fc2WriteRegister
FlyCapture2_C.h, 91
fc2WriteRegisterBlock
FlyCapture2_C.h, 92
fc2WriteRegisterBroadcast
FlyCapture2_C.h, 92
firmwareBuildTime
fc2CameraInfo, 9
firmwareVersion
fc2CameraInfo, 9
format
fc2Image, 22
frameCounter
fc2EmbeddedImageInfo, 14
frameRate
fc2AVIOption, 7
fc2H264Option, 21
fc2MJPGOption, 28

gain
fc2EmbeddedImageInfo, 14
gigEMajorVersion
fc2CameraInfo, 9
gigEMinorVersion
fc2CameraInfo, 9
gpuDescription
fc2SystemInfo, 32
grabMode
fc2Config, 11
grabTimeout
fc2Config, 11

height
fc2Format7ImageSettings, 15
fc2GigEImageSettings, 18
fc2H264Option, 21
hostPost
fc2GigEStreamChannel, 21

iidcVer
fc2CameraInfo, 10
imageHStepSize
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
imageImpl
fc2Image, 22
imageVStepSize
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
inputBitDepth
fc2LUTData, 27
interPacketDelay
fc2GigEStreamChannel, 21
interfaceType
fc2CameraInfo, 10
interlaced
fc2PNGOption, 29
ipAddress
fc2CameraInfo, 10
isColorCamera
fc2CameraInfo, 10
isReadable
fc2GigEProperty, 19
isWritable
fc2GigEProperty, 19
isochBusSpeed
fc2Config, 11

libraryList
fc2SystemInfo, 32

macAddress
fc2CameraInfo, 10
major
fc2Version, 37
manualSupported
fc2TriggerDelayInfo, 35
max
fc2GigEProperty, 19
fc2TriggerDelayInfo, 35
maxBytesPerPacket
fc2Format7PacketInfo, 16
maxHeight
fc2Format7Info, 16

fc2GigEImageSettingsInfo, 19
maxPacketSize
fc2Format7Info, 16
maxPacketsToResend
fc2GigEConfig, 17
maxValue
fc2StrobeInfo, 31
maxWidth
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
maximumBusSpeed
fc2CameraInfo, 10
microSeconds
fc2TimeStamp, 33
min
fc2GigEProperty, 19
fc2TriggerDelayInfo, 35
minNumImageNotifications
fc2Config, 11
minPacketSize
fc2Format7Info, 16
minValue
fc2StrobeInfo, 31
minor
fc2Version, 37
mode
fc2Format7ImageSettings, 15
fc2Format7Info, 16
fc2TriggerMode, 36
modeMask
fc2TriggerModeInfo, 37
modelName
fc2CameraInfo, 10

networkInterfaceIndex
fc2GigEStreamChannel, 21
nodeNumber
fc2CameraInfo, 10
nodeVendorId
fc2ConfigROM, 12
numBanks
fc2LUTData, 27
numBuffers
fc2Config, 11
numChannels
fc2LUTData, 27
numCpuCores
fc2SystemInfo, 32
numEntries
fc2LUTData, 27
numImageNotifications
fc2Config, 11

octets
fc2IPAddress, 25
fc2MACAddress, 27
offsetHStepSize
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
offsetVStepSize
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
offsetX
fc2Format7ImageSettings, 15
fc2GigEImageSettings, 18
offsetY
fc2Format7ImageSettings, 15
fc2GigEImageSettings, 18
onOff
fc2EmbeddedImageInfoProperty, 14
fc2StrobeControl, 30
fc2TriggerDelay, 34
fc2TriggerMode, 36
onOffSupported
fc2StrobeInfo, 31
fc2TriggerDelayInfo, 35
fc2TriggerModeInfo, 37
onePush
fc2TriggerDelay, 34
onePushSupported
fc2TriggerDelayInfo, 35
osDescription
fc2SystemInfo, 32
osType
fc2SystemInfo, 32
outputBitDepth
fc2LUTData, 27

pBusMgr
fc2InternalContext, 24
pCallback
fc2InternalImageCallback, 25
pCallbackData
fc2InternalImageCallback, 25
pCamera
fc2InternalContext, 24
pCameraControlDlg
fc2InternalGuiContext, 24
pCameraSelectionDlg
fc2InternalGuiContext, 24

pData
fc2Image, 22
pUnitAbbr
fc2TriggerDelayInfo, 35
pUnits
fc2TriggerDelayInfo, 35
packetSize
fc2Format7Info, 16
fc2GigEStreamChannel, 21
parameter
fc2TriggerMode, 36
pcieBusSpeed
fc2CameraInfo, 10
percentage
fc2Format7Info, 16
pixelFormat
fc2Format7ImageSettings, 15
fc2GigEImageSettings, 18
pixelFormatBitField
fc2Format7Info, 16
fc2GigEImageSettingsInfo, 19
polarity
fc2StrobeControl, 30
fc2TriggerMode, 36
polaritySupported
fc2StrobeInfo, 31
fc2TriggerModeInfo, 37
present
fc2StrobeInfo, 31
fc2TriggerDelay, 34
fc2TriggerDelayInfo, 35
fc2TriggerModeInfo, 37
progressive
fc2JPEGOption, 26
propType
fc2GigEProperty, 19
pszKeyword
fc2ConfigROM, 12

quality
fc2JPEGOption, 26
fc2JPG2Option, 26
fc2MJPGOption, 28

readOutSupported
fc2StrobeInfo, 31
fc2TriggerDelayInfo, 35
fc2TriggerModeInfo, 37
receivedDataSize
fc2Image, 22
recommendedBytesPerPacket
fc2Format7PacketInfo, 17
registerTimeout
fc2Config, 11
registerTimeoutRetries
fc2Config, 11
reserved
fc2AVIOption, 7
fc2CameraInfo, 10
fc2Config, 11
fc2ConfigROM, 12
fc2Format7ImageSettings, 15
fc2Format7Info, 16
fc2Format7PacketInfo, 17
fc2GigEConfig, 17
fc2GigEImageSettings, 18
fc2GigEImageSettingsInfo, 19
fc2GigEProperty, 20
fc2GigEStreamChannel, 21
fc2H264Option, 21
fc2ImageMetadata, 23
fc2JPEGOption, 26
fc2JPG2Option, 26
fc2LUTData, 27
fc2MJPGOption, 28
fc2PGMOption, 28
fc2PNGOption, 29
fc2PPMOption, 30
fc2StrobeControl, 30
fc2StrobeInfo, 31
fc2SystemInfo, 32
fc2TIFFOption, 33
fc2TimeStamp, 33
fc2TriggerDelay, 34
fc2TriggerDelayInfo, 35
fc2TriggerMode, 36
fc2TriggerModeInfo, 37
rows
fc2Image, 22

screenHeight
fc2SystemInfo, 32
screenWidth
fc2SystemInfo, 32
seconds
fc2TimeStamp, 33
sensorInfo
fc2CameraInfo, 10
sensorResolution
fc2CameraInfo, 10

serialNumber
    fc2CameraInfo, 10
shutter
    fc2EmbeddedImageInfo, 14
softwareTriggerSupported
    fc2TriggerModeInfo, 37
source
    fc2StrobeControl, 30
    fc2StrobeInfo, 31
    fc2TriggerMode, 36
sourceMask
    fc2TriggerModeInfo, 37
sourcePort
    fc2GigEStreamChannel, 21
stride
    fc2Image, 22
strobePattern
    fc2EmbeddedImageInfo, 14
subnetMask
    fc2CameraInfo, 10
supported
    fc2LUTData, 27
syncContext
    MultiSyncLibraryDefs_C.h, 116
syncCreateContext
    MultiSyncLibrary_C.h, 112
syncDestroyContext
    MultiSyncLibrary_C.h, 113
syncDisableCrossPCSynchronization
    MultiSyncLibrary_C.h, 113
syncEnableCrossPCSynchronization
    MultiSyncLibrary_C.h, 113
syncError
    MultiSyncLibraryDefs_C.h, 117
syncGetStatus
    MultiSyncLibrary_C.h, 113
syncGetTimeSinceSynced
    MultiSyncLibrary_C.h, 114
syncIsTimingBusConnected
    MultiSyncLibrary_C.h, 114
syncMessage
    MultiSyncLibraryDefs_C.h, 117
syncQueryCrossPCSynchronization-
        Setting
    MultiSyncLibrary_C.h, 114
syncRescanMasterTimingBus
    MultiSyncLibrary_C.h, 115
syncStart
    MultiSyncLibrary_C.h, 115
syncStop
    MultiSyncLibrary_C.h, 115
sysMemSize
    fc2SystemInfo, 32

timeoutForPacketResend
    fc2GigEConfig, 17
timestamp
    fc2EmbeddedImageInfo, 14
type
    fc2TriggerDelay, 34
    fc2TriggerDelayInfo, 35
    fc2Version, 37

unitBytesPerPacket
    fc2Format7PacketInfo, 17
unitSWVer
    fc2ConfigROM, 12
unitSpecId
    fc2ConfigROM, 12
unitSubSWVer
    fc2ConfigROM, 12
userDefinedName
    fc2CameraInfo, 10

value
    fc2GigEProperty, 20
    fc2PGRGuid, 29
valueA
    fc2TriggerDelay, 34
valueB
    fc2TriggerDelay, 34
valueReadable
    fc2TriggerModeInfo, 37
vendorName
    fc2CameraInfo, 10
vendorPixelFormatBitField
    fc2Format7Info, 16
    fc2GigEImageSettingsInfo, 19
vendorUniqueInfo_0
    fc2ConfigROM, 12
vendorUniqueInfo_1
    fc2ConfigROM, 12
vendorUniqueInfo_2
    fc2ConfigROM, 12
vendorUniqueInfo_3
    fc2ConfigROM, 12

whiteBalance
    fc2EmbeddedImageInfo, 14
width

fc2Format7ImageSettings, 15
fc2GigEImageSettings, 18
fc2H264Option, 21

xmlURL1
fc2CameraInfo, 10
xmlURL2
fc2CameraInfo, 10