SynView

.NET Reference Guide

Generated by Doxygen 1.8.7

Mon Nov 16 2015 14:19:47

Contents

Chapter 1

SynView .NET Reference Guide

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Namespace Index

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

Module Documentation

6.1 SynView .Net Class Library API

Modules

6.1.1 Detailed Description

6.2 SynView .Net Class Library Defines

Modules

- SynView .Net Class Library Status Definitions
- LvPixelFormat definitions

Namespaces

- NewElectronicTechnology
- NewElectronicTechnology::SynView

Typedefs

- typedef UInt32 SizeT
- typedef UInt32 LvFeature
- typedef Int32 LvStatus

6.2.1 Detailed Description

6.2.2 Typedef Documentation

6.2.2.1 typedef UInt32 LvFeature

Base typedef for the ID of the feature.

Note

Typedefs are not exposable in metadata, in the assembly API they are visible as base types, for example LvStatus is visible as Int32. But we use the typedefs in our documentation to make the API better readable.

6.2.2.2 typedef Int32 LvStatus

Error status.

Note

Typedefs are not exposable in metadata, in the assembly API they are visible as base types, for example LvStatus is visible as Int32. But we use the typedefs in our documentation to make the API better readable.

6.2.2.3 typedef UInt32 SizeT

Typedef equal to the size_t in standard library.

Note

Typedefs are not exposable in metadata, in the assembly API they are visible as base types, for example LvStatus is visible as Int32. But we use the typedefs in our documentation to make the API better readable.

6.3 SynView .Net Class Library Status Definitions

Typedefs

- typedef UInt32 LvEnum

6.3.1 Detailed Description

Most of the SynView methods return a number indicating the success or failure. The value 0 indicates success. A non-zero value indicates an error. You can use the LvLibrary::GetErrorMessage() to convert the number to a string with an error message. You can also use the LvLibrary::ThrowErrorEnable() to switch on/off the usage of exceptions, thrown upon a non-zero status returned from a method; then you do not to check every method call return value; instead you must catch the possible exceptions of the LvException type in your code. By default, exceptions are switched ON in the SynView .Net Class Library.

6.3.2 Typedef Documentation

6.3.2.1 typedef UInt32 LvEnum

Base typedef for the entry of the enumeration item.

Note

Typedefs are not exposable in metadata, in the assembly API they are visible as base types, for example LvStatus is visible as Int32. But we use the typedefs in our documentation to make the API better readable.

6.4 SynView .Net Class Library Enums

Modules

- SynView .Net Class Library Features
- SynView .Net Class Library Enumeration Entries

Enumerations

- enum LvStreamStartFlags : UInt32 { Default }
- enum LvStreamStopFlags : UInt32 { Default, Kill }
- enum LvUniLutFlags : UInt32 { None, HwLut }
- enum LvSaveFlag : UInt32 {
  None, RemoteFtr, LocalFtr, GenTlFtr,
  All, IgnoreVersion, IgnoreModel }
- enum LvLibInfo : LvEnum {
  BinPath, AppDataPath, UserDataPath, CfgPath,
  InstPath, IniFile, BuildDate }
- enum LvFtrGroup : LvEnum {
  DeviceRemote, SystemGtl, InterfaceGtl, DeviceGtl,
  StreamGtl, BufferGtl, SystemLocal, InterfaceLocal,
  DeviceLocal, StreamLocal, BufferLocal, RendererLocal,
  EventLocal, Buffer, Event, EventItemsGtl,
  BufferItemsGtl, SystemHidden, InterfaceHidden, DeviceHidden,
  StreamHidden, BufferHidden, RendererHidden, EventHidden }
- enum LvFtrType : LvEnum {
  Integer, Float, String, Enumeration,
  Boolean, Command, Category, StringList,
  Pointer, Buffer, Other }
- enum LvFtrGui : LvEnum {
  IntEdit, IntEditHex, IntSlider, IntSliderLog,
  FloatEdit, FloatSlider, FloatSliderLog, Label,
  StringEdit, CheckBox, ComboBox, Button,
  IpV4Address, IpMacAddress, Undefined }
- enum LvFtrVisibility : LvEnum { Beginner, Expert, Guru, Invisible }
- enum LvFtrAccess : LvEnum {
  NotImplemented, NotAvailable, WriteOnly, ReadOnly,
  ReadWrite }
- enum LvFtrInfo : LvEnum {
  IsStreamable, IsWrapped, IsSelector, IsCached,
  PollingTime, Name, DisplayName, Description,
  PhysicalUnits, ToolTip, SymbolicConst, SymbolicEnumConst,
  SelectedFeatures, SelectingFeatures, SymbolicGroupConst, ModuleName,
  FitsTo32Bit, TakeAsReadOnly, EnumEntryName, EnumEntryDisplayName,
  EnumEntryDescription, EnumEntryToolTip, EnumEntryAccess, EnumEntryValue,
  EnumEntryCount, EnumEntryNameMaxSize, InterfaceID, InterfaceDisplayName,
  InterfaceTlType, DeviceID, DeviceVendor, DeviceModel,
  DeviceTlType, DeviceDisplayName, DeviceAccessStatus }
- enum LvInfoDataType : LvEnum {
  Unknown, String, StringList, Int16,
  UInt16, Int32, UInt32, Int64,
  UInt64, Float64, Ptr, Bool,
  SizeT, Buffer }
- enum LvQueueOperation : LvEnum {
  InputToOutput, OutputDiscard, AllToInput, UnqueuedToInput,
  AllDiscard }

- enum LvEventType : LvEnum { Error, NewBuffer, FeatureDevEvent }
- enum LvEventDataInfo : LvEnum { Id, Value }
- enum LvRenderFlags : UInt32 { None, RepaintBackground, DontPaintIncomplete, IgnoreInvalidWinHandle }
- enum LvFindBy : LvEnum {
  UserID, VendorName, ModelName, TLType,
  DisplayName, GevIPAddress, GevMACAddress, SerialNumber,
  Any }

6.4.1 Detailed Description

6.4.2 Enumeration Type Documentation

6.4.2.1 enum LvEventDataInfo : LvEnum [strong]

LvEventDataInfo constants. Define values for the info specification in the LvEventGetDataInfo() function.

Enumerator

Id Represents the GenTL EVENT_DATA_ID - Event ID. See LvEventType for the explanation, what this ID means according to the event type.

Value Represents the GenTL EVENT_DATA_VALUE - Event Data. See LvEventType for the explanation, what this data means according to the event type.

6.4.2.2 enum LvEventType : LvEnum [strong]

LvEventType constants. Type of the event, for which the Event module is created - see LvEvent::Open(). Currently only the LvEventType::NewBuffer is supported by SynView.

Enumerator

Error Represents the GenTL EVENT_ERROR - Notification on module errors. For this type of event the LvEventDataInfo::Id is LvInfoDataType::Int32 and LvEventDataInfo::Value is LvInfoDataType::String.

NewBuffer Represents the GenTL EVENT_NEW_BUFFER - Notification on newly filled buffers placed to the output queue. For this type of event the LvEventDataInfo::Id is LvInfoDataType::Ptr (GenTL Buffer handle) and LvEventDataInfo::Value is LvInfoDataType::Ptr (Private pointer).

FeatureDevEvent Represents the GenTL EVENT_FEATURE_DEVEVENT. Notification if the GenTL Producer wants to inform the GenICam GenApi instance of the remote device that a GenApi compatible event was fired. This event is processed internally in SynView API - it is converted into the feature change callback - see the LvSystem::RegisterFeatureCallback() function. However, the thread which checks the GenTL event and converts it into the callbacks must be started explicitly by the application - see the LvEventStartThread() function. This event type can be opened only on the Device module.

6.4.2.3 enum LvFindBy : LvEnum [strong]

Enum values for the LvSystem::FindInterface() and LvInterface::FindDevice() functions.

Enumerator

UserID Can be used in the LvInterface::FindDevice() for finding the device by its User ID (nickname).

VendorName Can be used in the LvInterface::FindDevice() for finding the device by its vendor name.

ModelName Can be used in the LvInterface::FindDevice() for finding the device by its model name.

TLType Can be used in the LvSystem::FindInterface() or LvInterface::FindDevice() for finding the interface or device by its Transport Layer type. The search string can be then one of the following:

- "GEV" for GigE Vision,

- "CL" for Camera Link,
- "IIDC" for IIDC 1394,
- "UVC" for USB video class devices,
- "Custom" for not defined ones (for example the New Electronic Technology CorSight streaming device).

DisplayName Can be used in the LvSystem::FindInterface() or LvInterface::FindDevice() for finding the interface or device by its display name.

GevIPAddress Can be used in the LvSystem::FindInterface() or LvInterface::FindDevice() for finding the interface or device by its IP address (in case of the interface, it is the default IP address of the NIC).

GevMACAddress Can be used in the LvInterface::FindDevice() for finding the device by its model name.

SerialNumber Can be used in the LvInterface::FindDevice() for finding the device by its serial number.

Any Tries to find the string in all available IDs (UserID, VendorName, ModelName...).

6.4.2.4 enum LvFtrAccess : LvEnum [strong]

LvFtrAccess constants. Define the current feature access mode. Used in the LvGetAccess(). Also used for enumeration features in functions LvGetEnumValByStr() and LvGetEnumStrByVal().

Enumerator

NotImplemented The feature is not implemented at all.

NotAvailable The feature is implemented, but under the current conditions is not available.

WriteOnly The feature is available and is write only.

ReadOnly The feature is available and is read only.

ReadWrite The feature is available and is fully accessible.

6.4.2.5 enum LvFtrGroup : LvEnum [strong]

LvFtrGroup constants. Define the group of features. The group is composed of the module and the feature origin. The richest set is belonging to the Device module:

- Device remote features are those, which are provided by the device itself through GenICam GenApi.
- Device GenTL features are those, which are provided by the GenTL library through GenICam GenApi.
- Device local features are those, which are implemented directly in the SynView library. Used in LvGetNum↩ Features(), LvGetFeatureAt(), LvGetFeatureByName().

Enumerator

DeviceRemote Device remote features obtained from the device GenApi node tree.

SystemGtl System features obtained from the GenTL GenApi node tree.

InterfaceGtl Interface features obtained from the GenTL GenApi node tree.

DeviceGtl Device features obtained from the GenTL GenApi node tree.

StreamGtl Stream features obtained from the GenTL GenApi node tree.

BufferGtl Buffer features obtained from the GenTL GenApi node tree.

SystemLocal System local features, implemented in SynView.

InterfaceLocal Interface local features, implemented in SynView.

DeviceLocal Device local features, implemented in SynView.

StreamLocal Stream local features, implemented in SynView.

BufferLocal Buffer local features, implemented in SynView.

RendererLocal Renderer local features, implemented in SynView.

EventLocal Event local features, implemented in SynView.

Buffer Obsolete - will be removed. Buffer local features, implemented in SynView.

Event Obsolete - will be removed. Event local features, implemented in SynView.

EventItemsGtl Obsolete - will be removed. Event local GenTL features obtained from the GenTL plain C API.

BufferItemsGtl Obsolete - will be removed. Buffer local GenTL features obtained from the GenTL plain C API.

SystemHidden System hidden features. Do not use, reserved for special purposes.

InterfaceHidden Interface hidden features. Do not use, reserved for special purposes.

DeviceHidden Device hidden features. Do not use, reserved for special purposes.

StreamHidden Stream hidden features. Do not use, reserved for special purposes.

BufferHidden Buffer hidden features. Do not use, reserved for special purposes.

RendererHidden Renderer hidden features. Do not use, reserved for special purposes.

EventHidden Event hidden features. Do not use, reserved for special purposes.

6.4.2.6 enum LvFtrGui : LvEnum [strong]

LvFtrGui constants. Define the recommended GUI representation of the feature. Used in the LvGetType() function.

Enumerator

IntEdit The recommended representation is an edit box with a decimal value. Used by LvFtrType::Integer.

IntEditHex The recommended representation is an edit box with a hexadecimal value. Used by LvFtrType::Integer.

IntSlider The recommended representation is a linear slider. Used by LvFtrType::Integer.

IntSliderLog The recommended representation is a logarithmic slider. Used by LvFtrType::Integer.

FloatEdit The recommended representation is an edit box. Used by LvFtrType::Float.

FloatSlider The recommended representation is a linear slider. Used by LvFtrType::Float.

FloatSliderLog The recommended representation is a logarithmic slider. Used by LvFtrType::Float.

Label The recommended representation is read-only label. Used by LvFtrType::Category.

StringEdit The recommended representation is an edit box for a string. Used by LvFtrType::String.

CheckBox The recommended representation is a check box. Used by LvFtrType::Boolean.

ComboBox The recommended representation is a combo box. Used by LvFtrType::Boolean.

Button The recommended representation is a button. Used by LvFtrType::Command.

IpV4Address The recommended representation is an edit box for a string with an IP address in the form N.N.N.N. Used by LvFtrType::Integer.

IpMacAddress The recommended representation is an edit box for a string with a MAC address in the form XX:XX:XX:XX:XX:XX. Used by LvFtrType::Integer.

Undefined The recommended representation is not defined.

6.4.2.7 enum LvFtrInfo : LvEnum [strong]

LvFtrInfo constants. Define the info type when querying for feature info by the LvGetInfo() and LvGetInfoStr() functions.

Enumerator

IsStreamable Returns 1 if the feature has the Streamable attribute set. To be used in the LvGetInfo() function.

IsWrapped Returns 1 if the feature should not be used directly, because SynView provides for this functionality a native API. For example the AcquisitionStart and AcquisitionStop device remote features are wrapped by additional functionality in SynView (for example locking TL params before the AcquisitionStart command is issued). To be used in the LvGetInfo() function.

IsSelector Returns 1 if the feature is a selector, that means subsequent features are indexed by it . To be used in the LvGetInfo() function.

IsCached Returns 1 if the feature is cached. To be used in the LvGetInfo() function.

PollingTime Returns the polling time for a non-cached feature. If the feature is dependent on other non-cached features, the returned polling time is the minimum found. The polling time defines recommended time to update the non-cached feature. For example the LvDevice::DeviceTemperature is a typical non-cached feature - it changes independentely and a as it changes slowly, the recommended polling time might be 10000 = 10 seconds, i.e. the application, which displays the temperature, should update it on screen every 10 seconds. The returned value -1 means the polling time is not defined. To be used in the LvGetInfo() function.

Name Returns the feature Name. Do not confuse it with the DisplayName - the Name is the string identifier, by which the feature can be identified and a numeric ID can be obtained for further actions (generic feature access). To be used in the LvGetInfoStr() function.

DisplayName Returns the feature Display name for representation in GUI. To be used in the LvGetInfoStr() function.

Description Returns the feature Description text. To be used in the LvGetInfoStr() function.

PhysicalUnits Returns the feature Physical units, if defined. To be used in the LvGetInfoStr() function.

ToolTip Returns the feature Tooltip (a short description to be used in the GUI). To be used in the LvGetInfoStr() function.

SymbolicConst Returns the SynView symbolic constant of the feature, as a string (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

SymbolicEnumConst Returns the SynView symbolic constant of the enumeration feature, as a string (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

SelectedFeatures Returns the string ID of selected features belonging under this selector. Param = index (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

SelectingFeatures Returns the string ID of selecting features under which this feature belongs. Param = index (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

SymbolicGroupConst Returns the SynView symbolic constant for the feature group, to which the feature belongs, as a string (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

ModuleName Returns the string indicating the type of module, to which the fature belongs, for example "↩ System", "Interface", "Device", ... (utilized in the Source code generator). To be used in the LvGetInfoStr() function.

FitsTo32Bit Returns 1 if for this feature can be safely used 32-bit integer instead of 64-bit (if the feaure is of the LvFtrType::Integer type). This info is utilized in the source code generator. To be used in the LvGetInfo() function.

TakeAsReadOnly Returns 1 if this feature is either permanently read-only (cannot become read-write depending on other features), or the feature is writable, but it is not usual to set its value from code. This info is utilized in the source code generator. To be used in the LvGetInfo() function.

EnumEntryName Returns the symbolic name of the enum entry. To be used in the LvGetInfoStr() function. The Param specifies a zero based index of the entry or the SynView enum entry constant. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::EnumEntryCount parameter. If the Param is set to LV_ENUMENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryDisplayName Returns the display name of the enum entry. To be used in the LvGetInfoStr() function. The Param specifies a zero based index of the entry or the SynView enum entry constant. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::EnumEntryCount parameter. If the Param is set to LV_ENUMENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryDescription Returns the description of the enum entry. To be used in the LvGetInfoStr() function. The Param specifies a zero based index of the entry or the SynView enum entry constant. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::EnumEntryCount parameter. If the Param is set to LV_ENUMENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryToolTip Returns the tooltip of the enum entry. To be used in the LvGetInfoStr() function. The Param specifies a zero based index of the entry or the SynView enum entry constant. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::EnumEntryCount parameter. If the Param is set to LV_ENUMENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryAccess Returns the access of the enum entry (one of the LvFtrAccess constants). To be used in the LvGetInfo() function. The Param specifies a zero based index of the entry or the SynView enum entry constant. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::Enum↩EntryCount parameter. If the Param is set to LV_ENUMENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryValue Returns the SynView constant for the enum entry (if exists). To be used in the LvGetInfo() function. The Param specifies a zero based index of the entry. You can obtain the number of entries by the LvGetInfo() function with the LvFtrInfo::EnumEntryCount parameter. If the Param is set to LV_ENU↩MENTRY_CURRENT, the returned info is for the currently selected enum entry.

EnumEntryCount Returns the number of enum entries for the enum. To be used in the LvGetInfo() function.

EnumEntryNameMaxSize Returns the maximum string size needed (including terminating zero) for any entry name of the enum To be used in the LvGetInfo() function.

InterfaceID Returns the string ID of the interface. Param = interface index. This constant can be used only in the LvSystem module for enumerating unopened interfaces (LvGetInfoStr() function, as the Feature use LvSystem::Info).

InterfaceDisplayName Returns the Display name of the interface. Param = interface index. This constant can be used only in the LvSystem module for enumerating unopened interfaces (LvGetInfoStr() function, as the Feature use LvSystem::Info).

InterfaceTlType Returns the interface Transport layer type. Param = interface index. This constant can be used only in the LvSystem module for enumerating unopened interfaces (LvGetInfoStr() function, as the Feature use LvSystem::Info). For example a standard interface TL type is "GEV" for GigE-Vision devices.

DeviceID Returns the string ID of the device. Param = device index. This constant can be used only in the LvInterface module for enumerating unopened devices (LvGetInfoStr() function, as the Feature use LvInterface::Info).

DeviceVendor Returns the Vendor name of the device. Param = device index. This constant can be used only in the LvInterface module for enumerating unopened devices (LvGetInfoStr() function, as the Feature use LvInterface::Info).

DeviceModel Returns the Model name of the device. Param = device index. This constant can be used only in the LvInterface module for enumerating unopened devices (LvGetInfoStr() function, as the Feature use LvInterface::Info).

DeviceTlType Returns the Transport layer type of the device. Param = device index. This constant can be used only in the LvInterface module for enumerating unopened devices (LvGetInfoStr() function, as the Feature use LvInterface::Info).

DeviceDisplayName Returns the Display name the device. Param = device index. This constant can be used only in the LvInterface module for enumerating unopened devices (LvGetInfoStr() function, as the Feature use LvInterface::Info).

DeviceAccessStatus Returns the the device access. Param = device index. The returned value is one of the LvDeviceAccessStatus constants. Can be used only in the LvInterface module for enumerating unopened devices (LvGetInfo() function, as the Feature use LvInterface::Info).

6.4.2.8 enum LvFtrType : LvEnum [strong]

LvFtrType constants. Define the type of the feature. Used in the LvGetType() function.

Enumerator

Integer Integer type, use LvModule::GetInt32(), LvModule::SetInt32(), LvModule::GetInt64(), LvModule::SetInt64() to get/set a value.

Float Float type, use LvModule::GetFloat() and LvModule::SetFloat() to get/set a value.

String String type, use LvModule::GetString() and LvModule::SetString() to get/set a value.

Enumeration Enumeration type, use LvModule::GetEnum(), LvModule::SetEnum(), LvModule::GetEnumStr() and LvModule::SetEnumStr() to get/set a value.

Boolean Boolean type, use LvModule::GetInt32() and LvModule::SetInt32() to get/set a value.

Command Command type, use LvModule::CmdExecute() and LvModule::CmdIsDone() to execute and check.

Category Category type, used in the tree of feaures build.

StringList String list type (multiple strings in one, separated by terminating 0), use LvModule::GetString() and LvModule::SetString() to get/set a value.

Pointer Pointer type, use LvModule::GetPtr() and LvModule::SetPtr() to get/set a value.

Buffer Buffer type (in GenICam it corresponds with the Register type), use LvModule::GetBuffer() and LvModule::SetBuffer() to get/set a value. Do not confuse it with the LvBuffer module.

Other Unknown type, cannot be accessed.

6.4.2.9 enum LvFtrVisibility : LvEnum [strong]

LvFtrVisibility constants. Define the visibility level of the feature. Used in LvGetVisibility(). Should be used for displaying the feature tree (or list).

Enumerator

Beginner Beginner level - the feature should be displayed always.

Expert Expert level - the feature should be displayed if at least the Expert level is selected.

Guru Guru level - the feature should be displayed if at least the Guru level is selected.

Invisible Invisible - the feature should not be displayed.

6.4.2.10 enum LvInfoDataType : LvEnum [strong]

LvInfoDataType constants. The enum is used only by the LvEventGetDataInfo() function - this function follows the GenTL EventGetDataInfo() function, which uses different data types, than the GenApi.

Enumerator

Unknown Represents the GenTL INFO_DATATYPE_UNKNOWN info - Unknown data type.

String Represents the GenTL INFO_DATATYPE_STRING info - 0-terminated C string (ASCII encoded).

StringList Represents the GenTL INFO_DATATYPE_STRINGLIST info - Concatenated INFO_DATATYPE_STRING list. End of list is signaled with an additional 0.

Int16 Represents the GenTL INFO_DATATYPE_INT16 info - Signed 16 bit integer.

UInt16 Represents the GenTL INFO_DATATYPE_UINT16 info - unsigned 16 bit integer.

Int32 Represents the GenTL INFO_DATATYPE_INT32 info - signed 32 bit integer.

UInt32 Represents the GenTL INFO_DATATYPE_UINT32 info - unsigned 32 bit integer.

Int64 Represents the GenTL INFO_DATATYPE_INT64 info - signed 64 bit integer.

UInt64 Represents the GenTL INFO_DATATYPE_UINT64 info - unsingned64 bit integer.

Float64 Represents the GenTL INFO_DATATYPE_FLOAT64 info - Signed 64 bit floating point number.

Ptr Represents the GenTL INFO_DATATYPE_PTR info - Pointer type (void*). Size is platform dependent (32 bit on 32 bit platforms).

Bool Represents the GenTL INFO_DATATYPE_BOOL8 info - Boolean value occupying 8 bit. 0 for false and anything for true.

SizeT Represents the GenTL INFO_DATATYPE_SIZET info - Platform dependent unsigned integer (32 bit on 32 bit platforms).

Buffer Represents the GenTL INFO_DATATYPE_BUFFER info - Like the INFO_DATATYPE_STRING but with arbitrary data and no 0 termination.

6.4.2.11 enum LvLibInfo : LvEnum [strong]

Enum values for the Info parameter of the LvGetLibInfo(), LvGetLibInfoStr() and LvGetLibInfoStrSize() functions.

Enumerator

BinPath Returns the full path to the SynView binaries (applications and libraries - in Windows the Bin folder of SynView). LvFtrType::String.

AppDataPath Returns the full path to the SynView application data. This folder may be different from the BinPath, for example in Windows Vista the BinPath is write protected, while AppDataPath is at the read-write location and contains files like sv.synview.log etc. LvFtrType::String.

UserDataPath Returns the full path to the SynView user data. In Windows this is equal to AppDataPath. LvFtrType::String.

CfgPath Returns the full path to the SynView config data. In Windows this is equal to AppDataPath. LvFtrType::String.

InstPath Returns the full path to the SynView installation root folder. LvFtrType::String.

IniFile Returns the full path to the lv.synview.ini file. LvFtrType::String.

BuildDate Returns the build date of the library.

6.4.2.12 enum LvQueueOperation : LvEnum [strong]

LvQueueOperation constants. Define enum values for the LvStreamFlushQueue() function.

Enumerator

InputToOutput Represents the GenTL ACQ_QUEUE_INPUT_TO_OUTPUT. Flushes the input pool to the output queue and if necessary adds entries in the LvEventType::NewBuffer event data queue.

OutputDiscard Represents the GenTL ACQ_QUEUE_OUTPUT_DISCARD. Discards all buffers in the output queue and if necessary removes the entries from the event data queue.

AllToInput Represents the GenTL ACQ_QUEUE_ALL_TO_INPUT. Puts all buffers in the input pool. Even those in the output queue and discard entries in the event data queue.

UnqueuedToInput Represents the GenTL ACQ_QUEUE_UNQUEUED_TO_INPUT. Puts all buffers that are not in the input pool or the output queue in the input pool.

AllDiscard Represents the GenTL ACQ_QUEUE_ALL_DISCARD. Discards all buffers in the input pool and output queue.

6.4.2.13 enum LvRenderFlags : UInt32 [strong]

The flags passed as parameter to the functions LvRenderer::DisplayImage() and LvRenderer::Repaint().

Enumerator

None To be used when no flags are to be set.

RepaintBackground Before painting the image, the window background is repainted. This is done automaticaly whenever the change of the window size is detected, or display mode is switched. You can also call LvRenderer::DisplayImage() with 0 as the buffer handle and this flag just to erase image painting area.

DontPaintIncomplete If the buffer LvBufferFtr::IsIncomplete feature is true, it is not painted. The IsIncomplete feature indicates the contents of the buffer is a mixture of new and old image data, typically it happens when some packets with image data from a GigE camera are lost. If this flag is set simply the paint or repaint of such buffer is skipped, leaving whatever was before on the screen.

IgnoreInvalidWinHandle This flag has a meaning ony for the LvRenderer::CanDisplayImage() function. If used, this function will not return an error if the window handle was not yet assigned by the LvRenderer::SetWindow() function. This can be utilized for checking if the image is displayable before the display window is actually used.

6.4.2.14 enum LvSaveFlag : UInt32 [strong]

LvSaveFlag definitions Used in LvDevice::SaveSettings() and LvDevice::LoadSettings() functions.

Enumerator

None To be used when no flags are to be set.

RemoteFtr Save/load device remote XML features.

LocalFtr Save/load device local XML features.

GenTlFtr Save/load device GenTL XML features.

All Save/load device all features (combines all flags above).

IgnoreVersion If specified, the remote device FW version check is not done when reading the file - the file is read even if it was created by device with a different FW version (this may lead to errors by some features).

IgnoreModel If specified, the remote device model check is not done when reading the file - the file is read even if it was created by different device model (this may lead to errors by some features).

6.4.2.15 enum LvStreamStartFlags : UInt32 [strong]

LvStreamStart() flags definitions

Enumerator

Default Default stream start flag.

6.4.2.16 enum LvStreamStopFlags : UInt32 [strong]

LvStreamStop() flags definitions

Enumerator

Default Stop the acquisition engine when the currently running tasks like filling a buffer are completed. This is the default.

Kill Stop the acquisition engine immediately and leave buffers currently being filled in the Input Buffer Pool.

6.4.2.17 enum LvUniLutFlags : UInt32 [strong]

Flags definitions for LvDeviceUniSetLut() and LvDeviceUniGetLut() functions.

Enumerator

None To be used when no flags are to be set.

HwLut If present, the operation is done directly on HW LUT, passing the UniProcess mechanism.

6.5 SynView .Net Class Library Features

Enumerations

- enum LvSystemFtr : LvFeature {
  TLVendorName, TLModelName, TLID, TLVersion,
  TLPath, TLType, GenTLVersionMajor, GenTLVersionMinor,
  GevVersionMajor, GevVersionMinor, InterfaceUpdateList, InterfaceSelector,
  InterfaceID, GevInterfaceMACAddress, GevInterfaceDefaultIPAddress, GevInterfaceDefaultSubnetMask,
  GevInterfaceDefaultGateway, LvSystemDisplayName, Info }
- enum LvInterfaceFtr : LvFeature {
  InterfaceID, InterfaceType, GevInterfaceGatewaySelector, GevInterfaceGateway,
  GevMACAddress, GevInterfaceSubnetSelector, GevInterfaceSubnetIPAddress, GevInterfaceSubnetMask,
  DeviceUpdateList, DeviceSelector, DeviceID, DeviceVendorName,
  DeviceModelName, DeviceAccessStatus, GevDeviceIPAddress, GevDeviceSubnetMask,
  GevDeviceMACAddress, LvInterface_LvDeviceUserID, LvInterface_LvDeviceSerialNumber, LvInterface↩
  DisplayName,
  Info }
- enum LvDeviceFtr : LvFeature {
  DeviceVendorName, DeviceModelName, DeviceManufacturerInfo, DeviceVersion,
  DeviceFirmwareVersion, LvRecoveryFirmwareVersion, DeviceSerialNumber, DeviceUserID,
  LvSensorID, LvGrabberID, DeviceScanType, DeviceRegistersStreamingStart,
  DeviceRegistersStreamingEnd, DeviceRegistersCheck, DeviceRegistersValid, DeviceReset,
  DeviceClockSelector, DeviceClockFrequency, DeviceTemperatureSelector, DeviceTemperature,
  LvDeviceUpTime, LvDeviceType, SensorWidth, SensorHeight,
  WidthMax, HeightMax, Width, Height,
  OffsetX, OffsetY, PixelFormat, BinningHorizontal,
  BinningVertical, DecimationHorizontal, DecimationVertical, LvAOIMode,
  LvReadoutWidth, LvReadoutHeight, LvReadoutOffsetX, LvReadoutOffsetY,
  LvVariablePayloadSize, AcquisitionMode, TriggerSelector, TriggerMode,
  TriggerSoftware, TriggerSource, TriggerActivation, TriggerDelay,
  TriggerDivider, LvTriggerCaching, ExposureMode, LvLongRangeExposureMode,
  LvGlobalResetMode, ExposureTime, ExposureAuto, LvAcquisitionFrameRateControlMode,
  AcquisitionFrameRate, LineSelector, LineMode, LineFormat,
  LineSource, LineInverter, LineStatus, LineStatusAll,
  UserOutputSelector, UserOutputValue, UserOutputValueAll, UserOutputValueAllMask,
  CounterSelector, LvCounterMode, CounterEventSource, CounterReset,
  CounterValue, CounterDuration, TimerSelector, TimerDuration,
  TimerDelay, TimerTriggerSource, LvSpecialPurposeTriggerSelector, LvSpecialPurposeTriggerSource,
  LvSpecialPurposeTriggerActivation, LvSpecialPurposeTriggerSoftware, LvImageStampsResetMask, Lv↩
  ImageStampSelector,
  LvImageStampResetEnable, LvBootSwitch, LvBayerDecoderAlgorithm, LvBayerDecoderThreshold,
  LvWatchdogEnable, LvWatchdogTimerDuration, LvWatchdogTimerReset, LvWatchdogFailed,
  GainSelector, Gain, GainAuto, BlackLevelSelector,
  BlackLevel, BlackLevelAuto, ColorTransformationSelector, ColorTransformationEnable,
  ColorTransformationValueSelector, ColorTransformationValue, LvExternalDeviceControlMode, LvExternal↩
  ADCSelector,
  LvExternalADCValue, LvPowerSwitchCurrentAction, LvPowerSwitchSelector, LvPowerSwitchBoundADC,
  LvPowerSwitchDrive, LvPowerSwitchPulsePlus, LvPowerSwitchPulseMinus, LvLensControlCalibrate,
  LvLensControlMinusEnd, LvLensControlPlusEnd, LvLensControlPulsePeriod, LvLensControlDutyCycle,
  LvLensControlTargetApproach, LvLensControlNrSlowSteps, LvLensControlTargetPosition, LvLensControl↩
  AdjustPosition,
  LvPowerSwitchPulseDuration, LvLensControlMinCalibrationRange, LvLensControlCalibrateAll, LUTSelector,
  LUTEnable, LUTIndex, LUTValue, LUTValueAll,
  PayloadSize, GevVersionMajor, GevVersionMinor, GevDeviceModeIsBigEndian,
  GevDeviceModeCharacterSet, GevInterfaceSelector, GevMACAddress, GevSupportedOptionSelector,
  GevSupportedOption, GevCurrentIPConfigurationLLA, GevCurrentIPConfigurationDHCP, GevCurrentIP↩

ConfigurationPersistentIP,
GevCurrentIPAddress, GevCurrentSubnetMask, GevCurrentDefaultGateway, GevPersistentIPAddress,
GevPersistentSubnetMask, GevPersistentDefaultGateway, GevNumberOfInterfaces, GevMessageChannel↩
Count,
GevStreamChannelCount, GevHeartbeatTimeout, GevTimestampTickFrequency, GevTimestampControl↩
Latch,
GevTimestampControlReset, GevTimestampControlLatchReset, GevTimestampValue, GevCCP,
GevStreamChannelSelector, GevSCPInterfaceIndex, GevSCPHostPort, GevSCPSFireTestPacket,
GevSCPSDoNotFragment, GevSCPSBigEndian, GevSCPSPacketSize, GevSCPD,
GevSCDA, GevLinkSpeed, UserSetSelector, UserSetLoad,
UserSetSave, UserSetDefaultSelector, ChunkModeActive, ChunkSelector,
ChunkEnable, ChunkOffsetX, ChunkOffsetY, ChunkWidth,
ChunkHeight, ChunkPixelFormat, ChunkLinePitch, ChunkFrameID,
ChunkTimestamp, ChunkExposureTime, ChunkGainSelector, ChunkGain,
ChunkBlackLevel, ChunkLineStatusAll, ChunkLvExternalADCSelector, ChunkLvExternalADCValue,
EventSelector, EventNotification, LvSmartAppID, LvSmartAppInt1,
LvSmartAppInt2, LvSmartAppInt3, LvSmartAppInt4, LvSmartAppInt5,
LvSmartAppInt6, LvSmartAppInt7, LvSmartAppInt8, LvSmartAppInt9,
LvSmartAppInt10, LvSmartAppInt11, LvSmartAppInt12, LvSmartAppInt13,
LvSmartAppInt14, LvSmartAppInt15, LvSmartAppInt16, LvSmartAppInt17,
LvSmartAppInt18, LvSmartAppInt19, LvSmartAppInt20, LvSmartAppInt21,
LvSmartAppInt22, LvSmartAppInt23, LvSmartAppInt24, LvSmartAppInt25,
LvSmartAppInt26, LvSmartAppInt27, LvSmartAppInt28, LvSmartAppInt29,
LvSmartAppInt30, LvSmartAppInt31, LvSmartAppInt32, LvSmartAppUint1,
LvSmartAppUint2, LvSmartAppUint3, LvSmartAppUint4, LvSmartAppUint5,
LvSmartAppUint6, LvSmartAppUint7, LvSmartAppUint8, LvSmartAppUint9,
LvSmartAppUint10, LvSmartAppUint11, LvSmartAppUint12, LvSmartAppUint13,
LvSmartAppUint14, LvSmartAppUint15, LvSmartAppUint16, LvSmartAppUint17,
LvSmartAppUint18, LvSmartAppUint19, LvSmartAppUint20, LvSmartAppUint21,
LvSmartAppUint22, LvSmartAppUint23, LvSmartAppUint24, LvSmartAppUint25,
LvSmartAppUint26, LvSmartAppUint27, LvSmartAppUint28, LvSmartAppUint29,
LvSmartAppUint30, LvSmartAppUint31, LvSmartAppUint32, LvSmartAppAsciiCmdString,
LvSmartAppAsciiCmdExecute, LvSmartAppAsciiCmdFeedback, LvSmartAppAsciiCmdRetCode, LvSmart↩
AppPath,
LvSmartAppStart, EventLvLog, EventLvLogTimestamp, EventLvLogMessage,
EventLvSmartAppLog, EventLvSmartAppLogTimestamp, EventLvSmartAppLogMessage, LvSerialPort↩
BaudRate,
LvSerialPortParity, LvSerialPortDataBits, LvSerialPortStopBits, LvSerialPortTimeout,
LvSerialPortEOTMarker, LvSerialPortMaxResponseLength, LvSerialPortCommandString, LvSerialPort↩
CommandSend,
LvSerialPortCommandResponse, LvSerialPortCommandStatus, LvSmartAppExitEvent, LvWatchdogTimer↩
Value,
LvLensControlInvertedPolarity, GevMCPHostPort, GevMCDA, GevMCTT,
GevMCRC, ChunkLvSmartAppString, ChunkLvSmartAppIntSelector, ChunkLvSmartAppInt,
ChunkLvSmartAppUintSelector, ChunkLvSmartAppUint, ChunkLvSmartAppRegister, EventLvSmartApp↩
String,
EventLvSmartAppStringTimestamp, EventLvSmartAppStringValue, EventLvSmartAppInt, EventLvSmart↩
AppIntTimestamp,
EventLvSmartAppIntSelector, EventLvSmartAppIntValue, EventLvSmartAppUint, EventLvSmartAppUint↩
Timestamp,
EventLvSmartAppUintSelector, EventLvSmartAppUintValue, EventLvSmartAppRegister, EventLvSmart↩
AppRegisterTimestamp,
EventLvSmartAppRegisterValue, DeviceSFNCVersionMajor, DeviceSFNCVersionMinor, DeviceSFNC↩
VersionSubMinor,
LvLineDebounceDuration, ActionDeviceKey, ActionSelector, ActionGroupKey,
ActionGroupMask, LvLensControlCalibrationStatus, LvLUTMode, BalanceRatioSelector,
BalanceRatio, BalanceWhiteAuto, GevDeviceClass, GevIPConfigurationStatus,
GevDiscoveryAckDelay, GevGVCPExtendedStatusCodes, GevGVCPPendingAck, GevGVCPHeartbeat↩

Disable,
GevGVCPPendingTimeout, GevPrimaryApplicationSwitchoverKey, GevPrimaryApplicationSocket, GevPrimaryApplicationIPAddress,
GevMCSP, GevSCCFGUnconditionalStreaming, GevSCCFGExtendedChunkData, GevSCPDirection,
GevSCSP, ChunkLvTriggerDelayed, EventLvTriggerDropped, EventLvTriggerDroppedTimestamp,
LvStrobeEnable, LvStrobeDurationMode, LvStrobeDuration, LvStrobeDelay,
LvStrobeBrightness, LvStrobeDropMode, LvLUTReset, ChunkLvStrobeDropped,
ReverseX, ReverseY, RegionSelector, RegionMode,
RegionDestination, AcquisitionFrameCount, AcquisitionBurstFrameCount, LvCustomID,
LvCustomInfo, LvCustomRegMode, LvCustomRegAddr, LvCustomRegData,
LvCustomRegMux, LinePitch, ChunkLvFrameAbort, ChunkLvTriggerDropped,
ChunkLvTriggerError, ChunkLvEncoderPosition, ChunkLvEncoderRotation, DeviceID,
DeviceType, GevDeviceIPAddress, GevDeviceSubnetMask, GevDeviceMACAddress,
GevDeviceGateway, LvGevDeviceStreamCaptureMode, StreamSelector, StreamID,
DeviceEndianessMechanism, LvGevFindMaxPacketSize, LvGevPacketSizeValue, LvGevTestPacketSize,
LvGevPacketSizeTestSuccess, LvGevCCTT, LvGevCCRC, LvCCStatus,
LvDeviceDisplayName, LvDeviceIsAcquiring, LvUniProcessMode, LvUniProcessEnableInPlace,
LvUniPixelFormat, LvUniProcessPayloadSize, LvUniLinePitch, LvUniBayerDecoderAlgorithm,
LvUniBrightness, LvUniContrast, LvUniGamma, LvUniBalanceRatioSelector,
LvUniBalanceRatio, LvUniBalanceWhiteAuto, LvUniBalanceWhiteReset, LvUniColorTransformationSelector,
LvUniColorTransformationEnable, LvUniColorTransformationValueSelector, LvUniColorTransformationValue,
LvUniSaturation,
LvUniProcessExecution, LvUniLUTMode, LvUniLUTSelector, LvUniLUTEnable,
LvUniLUTIndex, LvUniLUTValue, LvUniLUTValueAll, LvUniColorTransformationMode,
Info }

- enum LvStreamFtr : LvFeature {
  StreamID, StreamAnnouncedBufferCount, StreamAcquisitionModeSelector, StreamAnnounceBufferMinimum,
  StreamType, LvStreamDisplayName, LvCalcPayloadSize, LvPostponeQueueBuffers,
  LvAwaitDeliveryLimit, LvAutoAllocateProcessBuffers, LvPreallocateProcessBuffers, LvNumDelivered,
  LvNumUnderrun, LvNumAnnounced, LvNumQueued, LvNumAwaitDelivery,
  LvIsGrabbing, LvNumAborted, LvNumStarted, Info }
- enum LvRendererFtr : LvFeature {
  LvAutoDisplay, LvRenderType, LvOffsetX, LvOffsetY,
  LvWidth, LvHeight, LvIgnoreAspectRatio, LvDisableScaleUp,
  LvDisableScaleDown, LvCenterImage, LvNumberOfTiles, LvColumns,
  LvRows, LvTileGap, LvAutoTileCalculation, Info }
- enum LvEventFtr : LvFeature { EventType, NumInQueue, NumFired }
- enum LvBufferFtr : LvFeature {
  Base, Size, UserPtr, TimeStamp,
  NewData, IsQueued, IsAcquiring, IsIncomplete,
  TIType, SizeFilled, Width, Height,
  XOffset, YOffset, XPadding, YPadding,
  FrameId, ImagePresent, ImageOffset, PayloadType,
  PixelFormat, PixelFormatNameSpace, DeliveredImageHeight, DeliveredChunkPayloadSize,
  ChunkLayoutId, FileName, UniBase, UniSize,
  ProcessBase, ProcessSize, ExecProcess, UniImageOffset }

6.5.1 Detailed Description

6.5.2 Enumeration Type Documentation

6.5.2.1 enum LvBufferFtr : LvFeature [strong]

LvBufferFtr constants.

Enumerator

Base Represents the GenTL BUFFER_INFO_BASE info - Base address of the buffer memory. LvFtrType::↩ Pointer.

Size Represents the GenTL BUFFER_INFO_SIZE info - Size of the buffer in bytes. LvFtrType::Integer.

UserPtr Represents the GenTL BUFFER_INFO_USER_PTR info - The user pointer (supplied by the application when the buffer was allocated). LvFtrType::Pointer. Important: This pointer MUST NOT be used in the C++ API and .Net Class Library, where this pointer is utilized internally for the LvBuffer class instance. The actual User pointer is available by the LvBuffer::GetUserPtr() function.

TimeStamp Represents the GenTL BUFFER_INFO_TIMESTAMP info - Timestamp the buffer was acquired. The unit is device/implementation dependent. LvFtrType::Integer.

NewData Represents the GenTL BUFFER_INFO_NEW_DATA info - Flag to indicate that the buffer contains new data since the last call. LvFtrType::Boolean.

IsQueued Represents the GenTL BUFFER_INFO_ISQUEUED info - Flag to indicate if the buffer is in the input pool or output queue. LvFtrType::Boolean.

IsAcquiring Represents the GenTL BUFFER_INFO_ISACQUIRING info - Flag to indicate that the buffer is currently being filled with data. LvFtrType::Boolean.

IsIncomplete Represents the GenTL BUFFER_INFO_ISINCOMPLETE info - Flag to indicate that a buffer was filled, but an error occurred during that process. LvFtrType::Boolean.

TlType Represents the GenTL BUFFER_INFO_TLTYPE info - Transport layer technologies that are supported. LvFtrType::String.

SizeFilled Represents the GenTL BUFFER_INFO_SIZE_FILLED info - Number of bytes written into the buffer last time it has been filled. This value is reset to 0 when the buffer is placed into the Input Buffer Pool. LvFtrType::Integer.

Width Represents the GenTL 1.2 BUFFER_INFO_WIDTH info - Width of the data in the buffer in number of pixels. This information refers for example to the width entry in the GigE Vision image stream data leader. LvFtrType::Integer.

Height Represents the GenTL 1.2 BUFFER_INFO_HEIGHT info - Height of the data in the buffer in number of pixels as configured. For variable size images this is the max Height of the buffer. For example this information refers to the height entry in the GigE Vision image stream data leader. LvFtrType::Integer.

XOffset Represents the GenTL 1.2 BUFFER_INFO_XOFFSET info - XOffset of the data in the buffer in number of pixels from the image origin to handle areas of interest. This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Integer.

YOffset Represents the GenTL 1.2 BUFFER_INFO_YOFFSET info - YOffset of the data in the buffer in number of lines from the image origin to handle areas of interest. This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Integer.

XPadding Represents the GenTL 1.2 BUFFER_INFO_XPADDING info - XPadding of the data in the buffer in number of bytes. This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Integer.

YPadding Represents the GenTL 1.2 BUFFER_INFO_YPADDING info - YPadding of the data in the buffer in number of bytes. This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Integer.

FrameId Represents the GenTL 1.2 BUFFER_INFO_FRAMEID info - A sequentially incremented number of the frame. This information refers for example to the information provided in the GigE Vision image stream block id. The wrap around of this number is transportation technology dependent. For GigE Vision it is (so far) 16bit wrapping to 1. LvFtrType::Integer.

ImagePresent Represents the GenTL 1.2 BUFFER_INFO_IMAGEPRESENT info - Flag to indicate if the current data in the buffer contains image data. This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Boolean.

ImageOffset Represents the GenTL 1.2 BUFFER_INFO_IMAGEOFFSET info - Offset of the image data from the beginning of the delivered buffer in bytes. Applies for example when delivering the image as part of chunk data or on technologies requiring specific buffer alignment. LvFtrType::Integer.

PayloadType Represents the GenTL 1.2 BUFFER_INFO_PAYLOADTYPE info - Payload type of the data. This information refers to the constants defined in GenTL PAYLOADTYPE_IDs (UNKNOWN=0, IMAG↩ E=1, RAW_DATA=2, FILE=3, CHUNK_DATA=4, CUSTOM=1000) LvFtrType::Integer.

PixelFormat Represents the GenTL 1.2 BUFFER_INFO_PIXELFORMAT info - This information refers for example to the information provided in the GigE Vision image stream data leader. The interpretation of the pixel format depends on the namespace the pixel format belongs to. This can be inquired using the PixelFormatNameSpace feature. LvFtrType::Integer.

PixelFormatNameSpace Represents the GenTL 1.2 BUFFER_INFO_PIXELFORMAT_NAMESPACE info - This information refers to the constants defined in GenTL 1.2 PIXELFORMAT_NAMESPACE_IDs to allow interpretation of PixelFormat (UNKNOWN=0, GEV=1, IIDC=2, CUSTOM=1000). LvFtrType::Integer.

DeliveredImageHeight Represents the GenTL 1.2 BUFFER_INFO_DELIVERED_IMAGEHEIGHT info - The number of lines in the current buffer as delivered by the transport mechanism. For area scan type images this is usually the number of lines configured in the device. For variable size linescan images this number may be lower than the configured image height. This information refers for example to the information provided in the GigE Vision image stream data trailer. LvFtrType::Integer.

DeliveredChunkPayloadSize Represents the GenTL 1.2 BUFFER_INFO_DELIVERED_CHUNKPAYLOA↩ DSIZE info - This information refers for example to the information provided in the GigE Vision image stream data leader. LvFtrType::Integer.

ChunkLayoutId Represents the GenTL 1.2 BUFFER_INFO_CHUNKLAYOUTID info - This information refers for example to the information provided in the GigE Vision image stream data leader. The chunk layout id serves as an indicator that the chunk layout has changed and the application should re-parse the chunk layout in the buffer. When a chunk layout (availability or position of individual chunks) changes since the last buffer delivered by the device through the same stream, the device MUST change the chunk layout id. As long as the chunk layout remains stable, the camera MUST keep the chunk layout id intact. When switching back to a layout, which was already used before, the camera can use the same id again or use a new id. A chunk layout id value of 0 is invalid. It is reserved for use by cameras not supporting the layout id functionality. LvFtrType::Integer.

FileName Represents the GenTL 1.2 BUFFER_INFO_FILENAME info - This information refers for example to the information provided in the GigE Vision image stream data leader. For other technologies this is to be implemented accordingly. Since this is GigE Vision related information and the filename in GigE Vision is UTF8 coded, this filename is also UTF8 coded. LvFtrType::Integer.

UniBase Unified base address of the buffer. If the image was processed to the output buffer, the pointer to the output buffer is returned, otherwise the pointer to the acquisition buffer is returned. This enables to write simple universal code for image handling. LvFtrType::Pointer. SynView feature.

UniSize Size of the buffer returned on LvBuffer::UniBase call. LvFtrType::Integer. SynView feature.

ProcessBase Pointer to the process buffer, attached to this acquisition buffer. LvFtrType::Pointer. SynView feature.

ProcessSize Size of the process buffer, attached to this acquisition buffer. LvFtrType::Integer. SynView feature.

ExecProcess Executes the SW image processing of the buffer. To be used when the LvDevice::LvUni↩ ProcessExecution is set to LvUniProcessExecution::OnExplicitRequest. LvFtrType::Command. SynView feature.

UniImageOffset Unified image offset. If the image was processed to the output buffer, the image offset to the output buffer is returned, otherwise the image offset to the acquisition buffer is returned. This enables to write simple universal code for image handling. LvFtrType::Integer. SynView feature.

6.5.2.2 enum LvDeviceFtr : LvFeature [strong]

LvDeviceFtr constants.

Enumerator

DeviceVendorName Name of the manufacturer of the device. LvFtrType::String. Device remote feature (Lv↩ FtrGroup::DeviceRemote).

DeviceModelName Model name of the device. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceManufacturerInfo Manufacturer information about the device. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceVersion Version of the device. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceFirmwareVersion Version of the firmware loaded in the device. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvRecoveryFirmwareVersion String that indicates the version of the firmware and software to which the device would recover. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceSerialNumber Device identifier (serial number). LvFtrType::String. Note: This feature is called DeviceID in the SFNC, but we use rather the DeviceSerialNumber in order not to confuse it with the GenTL DeviceID, which is used for the device opening. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceUserID User-programmable device identifier. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSensorID Serial number of the sensor board. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvGrabberID Serial number of the grabber board. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceScanType Scan type of the sensor. LvFtrType::Enumeration. Values: LvDeviceScanType. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceRegistersStreamingStart Prepare the device for registers streaming without checking for consistency. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceRegistersStreamingEnd Announce the end of registers streaming. This will do a register set validation for consistency and activate it. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceRegistersCheck Perform the validation of the current register set for consistency. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceRegistersValid Reports if the current register set is valid and consistent. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceReset Resets the device and to put it in its power up state. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceClockSelector Selects a device clock frequency to be configured. LvFtrType::Enumeration. Values: LvDeviceClockSelector. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceClockFrequency Frequency of the selected clock in Hz. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceTemperatureSelector Selects the location within the device, where the temperature will be measured. LvFtrType::Enumeration. Values: LvDeviceTemperatureSelector. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceTemperature Current temperature at the selected location in degrees of Celcius LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvDeviceUpTime Current up-time of the device in milliseconds. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvDeviceType String that indicates the basic type of the device. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

SensorWidth Effective width of the sensor in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

SensorHeight Effective height of the sensor in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

WidthMax Maximum width of the image in pixels. The dimension is calculated after applying horizontal binning, decimation or readout width. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

HeightMax Maximum height of the image in pixels. The dimension is calculated after applying vertical binning, decimation or readout height. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

Width Image width provided by the device in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

Height Image height provided by the device in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

OffsetX Horizontal offset from the origin of the AOI (area of interest) in pixels. The AOI is applied to the result of binning and or decimation. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

OffsetY Vertical offset from the origin of the AOI (area of interest) in pixels. The AOI is applied to the result of binning and or decimation. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

PixelFormat Pixel format provided by the device. The feature combines pixel coding, size and color filter attributes. LvFtrType::Enumeration. Values: see LvPixelFormat. Device remote feature (LvFtrGroup::DeviceRemote).

BinningHorizontal Horizontal binning, number of horizontal pixels to combine together. This increases the intensity (and S/N ratio) of the pixels and reduces the horizontal resolution (width) of the image. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

BinningVertical Vertical binning, number of vertical pixels to combine together. This increases the intensity (and S/N ratio) of the pixels and reduces the vertical resolution (height) of the image. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

DecimationHorizontal Horizontal decimation (sub-sampling) of the image. This reduces the horizontal resolution (width) of the image by the specified factor. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

DecimationVertical Vertical decimation (sub-sampling) of the image. This reduces the vertical resolution (height) of the image by the specified factor. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvAOIMode Selects the mode of controlling the area of interest LvFtrType::Enumeration. Values: LvAOIMode. Device remote feature (LvFtrGroup::DeviceRemote).

LvReadoutWidth Width of the sensor-side area of interest in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvReadoutHeight Height of the sensor-side area of interest in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvReadoutOffsetX X offset (left offset) for the sensor-side area of interest in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvReadoutOffsetY Y offset (top offset) for the sensor-side area of interest in pixels. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvVariablePayloadSize This flag controls, whether the payload size can change during acquisition. When set, the image dimensions and other paramenters can vary during acquisition. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

AcquisitionMode Sets the acquisition mode of the device. It defines mainly the number of frames to capture during an acquisition and the way the acquisition stops. LvFtrType::Enumeration. Values: LvAcquisitionMode. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerSelector Selects the type of trigger to configure. LvFtrType::Enumeration. Values: LvTriggerSelector. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerMode Controls if the selected trigger is active. LvFtrType::Enumeration. Values: LvTriggerMode. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerSoftware Generates a software trigger when trigger source is set to 'software' or any physical line. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerSource Specifies the internal signal or physical input line to use as the trigger source. LvFtrType::Enumeration. Values: LvTriggerSource. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerActivation Activation mode of the trigger - specifies which edge of the signal is active. LvFtrType::↩ Enumeration. Values: LvTriggerActivation. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerDelay Trigger delay in microseconds, specifies a delay introduced between the trigger reception and its actual activation. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

TriggerDivider Used to divide the number of incoming trigger pulses by an integer factor. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvTriggerCaching Sets the caching mode for the selected trigger. The feature controls how early triggers are treated by the device. LvFtrType::Enumeration. Values: LvTriggerCaching. Device remote feature (LvFtrGroup::DeviceRemote).

ExposureMode Controls the exposure (shutter) mode applied for each acquisition. LvFtrType::Enumeration. Values: LvExposureMode. Device remote feature (LvFtrGroup::DeviceRemote).

LvLongRangeExposureMode Switches to mode with wider range of exposure times, but slightly higher jitter. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

LvGlobalResetMode Switches to mode with wider range of exposure times, but slightly higher jitter. LvFtr↩ Type::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ExposureTime Exposure time in microseconds. The feature controls how long are the pixels exposed to illumination. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

ExposureAuto Selects the automatic exposure mode. LvFtrType::Enumeration. Values: LvExposureAuto. Device remote feature (LvFtrGroup::DeviceRemote).

LvAcquisitionFrameRateControlMode Switches the acquisition frame rate control on or off. The camera might internally switch to different working mode, which can decrease the maximum frame rate. LvFtr↩ Type::Enumeration. Values: LvAcquisitionFrameRateControlMode. Device remote feature (LvFtrGroup↩ ::DeviceRemote).

AcquisitionFrameRate Acquisition frame rate in frames per second (Hz) - the frequency with which the image frames are captured. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LineSelector Selects the I/O line for querying and configuration. Note that to use given line to drive a device feature (trigger, counter, etc.), source of the given feature has to refer to the line. LvFtrType::Enumeration. Values: LvLineSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LineMode Line mode - controls, whether given line is used as signal input or output. LvFtrType::Enumeration. Values: LvLineMode. Device remote feature (LvFtrGroup::DeviceRemote).

LineFormat This feature controls the current electrical format of the selected physical input or output Line. LvFtrType::Enumeration. Values: LvLineFormat. Device remote feature (LvFtrGroup::DeviceRemote).

LineSource Selects a device internal signal that should drive the output signal of the selected line. LineMode must be Output. Not applicable for input lines. LvFtrType::Enumeration. Values: LvLineSource. Device remote feature (LvFtrGroup::DeviceRemote).

LineInverter Inverts the signal output on the selected line. LvFtrType::Boolean. Device remote feature (Lv↩ FtrGroup::DeviceRemote).

LineStatus Reports the current status of the selected line. LvFtrType::Boolean. Device remote feature (Lv↩ FtrGroup::DeviceRemote).

LineStatusAll Bit field indicating status of all i/o lines. LvFtrType::Integer. Device remote feature (LvFtr↩ Group::DeviceRemote).

UserOutputSelector Selects the user output for querying and configuration. LvFtrType::Enumeration. Values: LvUserOutputSelector. Device remote feature (LvFtrGroup::DeviceRemote).

UserOutputValue Reports the current status of the selected user output. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

UserOutputValueAll Bit field indicating status of all user outputs. Only the bits defined in the User Output Value All Mask are used, the others are ignored. LvFtrType::Integer. Device remote feature (LvFtrGroup↩ ::DeviceRemote).

UserOutputValueAllMask Mask for the User Output Value All bitfield - defines which bits are used to change a user output value and which are ignored. LvFtrType::Integer. Device remote feature (LvFtrGroup::↩ DeviceRemote).

CounterSelector Selects which counter to configure. LvFtrType::Enumeration. Values: LvCounterSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LvCounterMode Selects working mode of the selected counter. LvFtrType::Enumeration. Values: LvCounterMode. Device remote feature (LvFtrGroup::DeviceRemote).

CounterEventSource Internal device signal incrementing the selected counter. LvFtrType::Enumeration. Values: LvCounterEventSource. Device remote feature (LvFtrGroup::DeviceRemote).

CounterReset This command resets the selected counter LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

CounterValue Reads or sets the current value of the selected counter. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

CounterDuration Duration (or number of events) before the counter end event is generated and the counter expires. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

TimerSelector Selects which timer to configure. LvFtrType::Enumeration. Values: LvTimerSelector. Device remote feature (LvFtrGroup::DeviceRemote).

TimerDuration Sets the duration (in microseconds) of the timer active pulse. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

TimerDelay Sets the delay (in microseconds) applied between activating the timer and issuing the timer active signal. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

TimerTriggerSource Internal device signal activating the selected timer. LvFtrType::Enumeration. Values: LvTimerTriggerSource. Device remote feature (LvFtrGroup::DeviceRemote).

LvSpecialPurposeTriggerSelector Selects the special purpose trigger type to configure. LvFtrType::Enumeration. Values: LvSpecialPurposeTriggerSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LvSpecialPurposeTriggerSource Specifies the internal signal or physical input line to use as the trigger source. LvFtrType::Enumeration. Values: LvSpecialPurposeTriggerSource. Device remote feature (LvFtrGroup::DeviceRemote).

LvSpecialPurposeTriggerActivation Activation mode of the trigger - specifies which edge of the signal is active. LvFtrType::Enumeration. Values: LvSpecialPurposeTriggerActivation. Device remote feature (LvFtrGroup::DeviceRemote).

LvSpecialPurposeTriggerSoftware Generates a software trigger for the selected trigger action when trigger source is set to 'software' or any physical line. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvImageStampsResetMask A single bitfield that selects which features will be reset by the timestamp reset trigger in one access. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvImageStampSelector Selects an image stamp type for configuration. LvFtrType::Enumeration. Values: LvImageStampSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LvImageStampResetEnable Enables/disables the reset trigger functionality for the selected image stamp type. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

LvBootSwitch Selects the firmware type to load on next boot. LvFtrType::Enumeration. Values: LvBootSwitch. Device remote feature (LvFtrGroup::DeviceRemote).

LvBayerDecoderAlgorithm Selects the algorithm used by the Bayer decoder. LvFtrType::Enumeration. Values: LvBayerDecoderAlgorithm. Device remote feature (LvFtrGroup::DeviceRemote).

LvBayerDecoderThreshold Sets the threshold controlling the performance of the variable gradient Bayer decoder algorithm. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvWatchdogEnable Enables the watchdog reset function. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

LvWatchdogTimerDuration Watchdog timeout - when watchdog is enabled, the device reboots when the timeout specified expires. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvWatchdogTimerReset Resets the watchdog timer, the watchdog starts counting the specified timeout again. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvWatchdogFailed Signals that the last device reboot was initiated by the watchdog function. After reading, reset this flag explicitly, it wouldn't be affected by a 'warm' system reboot. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GainSelector Selects which gain type to configure. LvFtrType::Enumeration. Values: LvGainSelector. Device remote feature (LvFtrGroup::DeviceRemote).

Gain Gain value for the selected gain type in dB. This is an amplification factor applied to the video signal. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

GainAuto Controls the automatic gain control (AGC) mode. LvFtrType::Enumeration. Values: LvGainAuto. Device remote feature (LvFtrGroup::DeviceRemote).

BlackLevelSelector Selects which black level type to configure. LvFtrType::Enumeration. Values: LvBlackLevelSelector. Device remote feature (LvFtrGroup::DeviceRemote).

BlackLevel Controls the analog black level. This represents a DC offset applied to the video signal. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

BlackLevelAuto Controls the automatic black level mode. LvFtrType::Enumeration. Values: LvBlackLevelAuto. Device remote feature (LvFtrGroup::DeviceRemote).

ColorTransformationSelector Selects which color transformation module is controlled by the color transformation features. It also gives particular meaning to individual color transformation gains. LvFtrType::Enumeration. Values: LvColorTransformationSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ColorTransformationEnable Activates the selected Color Transformation module. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ColorTransformationValueSelector Selects the gain factor or offset of the transformation matrix to configure LvFtrType::Enumeration. Values: LvColorTransformationValueSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ColorTransformationValue Value of the selected color transformation matrix entry. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvExternalDeviceControlMode Selects the operation mode of external device control. LvFtrType::Enumeration. Values: LvExternalDeviceControlMode. Device remote feature (LvFtrGroup::DeviceRemote).

LvExternalADCSelector Selects the external ADC to configure. LvFtrType::Enumeration. Values: LvExternalADCSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LvExternalADCValue Reads the value of the selected external ADC. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchCurrentAction Reports the automated action currently performed by a power switch. LvFtrType::Enumeration. Values: LvPowerSwitchCurrentAction. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchSelector Selects the power switch to configure. LvFtrType::Enumeration. Values: LvPowerSwitchSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchBoundADC Sets an external ADC to the selected power switch. The bound pair will work together during the automatic operation. LvFtrType::Enumeration. Values: LvPowerSwitchBoundADC. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchDrive Drives the selected power switch with desired polarity. LvFtrType::Enumeration. Values: LvPowerSwitchDrive. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchPulsePlus Pulses the selected power switch with plus polarity. Available in the automatic operation mode. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchPulseMinus Pulses the selected power switch with minus polarity. Available in the automatic operation mode. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlCalibrate Starts an automatic calibration on the selected power switch and bounded ADCs. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlMinusEnd Represents the calibrated ADC value corresponding with the power switch's minus end. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlPlusEnd Represents the calibrated ADC value corresponding with the power switch's plus end. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlPulsePeriod Represents the calibrated slow motion pulse period for the selected power switch, in microsesonds. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlDutyCycle Represents the calibrated slow motion duty cycle for the selected power switch (in %). Defines how much of the pulse period is the power switch actually active. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlTargetApproach Selects how the target lens position should be approached. LvFtrType::Enumeration. Values: LvLensControlTargetApproach. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlNrSlowSteps Sets the number of slow steps required before reaching the target position LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlTargetPosition Sets the target position (value) of the ADC bound to the selected power switch LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlAdjustPosition Adjusts the required target position (value) of the ADC bound to the selected power switch. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvPowerSwitchPulseDuration Duration (in microseconds) of the pulses issued at the power switches. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlMinCalibrationRange Minimum value range that has to be reached on the external ADC to count the calibration as valid. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlCalibrateAll Starts an automatic calibration on the active power switches and bounded ADCs. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LUTSelector Selects which LUT to configure. LvFtrType::Enumeration. Values: LvLUTSelector. Device remote feature (LvFtrGroup::DeviceRemote).

LUTEnable Activates the selected LUT. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

LUTIndex Index of the element to access in the selected LUT LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LUTValue Value of the element for the current index in the selected LUT. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LUTValueAll This register accesses the entire content of the selected LUT in one chunk access. LvFtrType::Buffer. Device remote feature (LvFtrGroup::DeviceRemote).

PayloadSize Provides the number of bytes transferred for each image by the device, including image and chunk data. The value defines the required size of the target buffer used for acquisition. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevVersionMajor Major version of the GigE Vision specification implemented by the device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevVersionMinor Minor version of the GigE Vision specification implemented by the device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevDeviceModeIsBigEndian Endianess of the device registers. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevDeviceModeCharacterSet Character set used by all the strings of the device registers. LvFtrType::Enumeration. Values: LvGevDeviceModeCharacterSet. Device remote feature (LvFtrGroup::DeviceRemote).

GevInterfaceSelector Selects which physical network interface to control. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMACAddress MAC address of the network interface. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSupportedOptionSelector Selects the GEV option to interrogate for existing support. LvFtrType::Enumeration. Values: LvGevSupportedOptionSelector. Device remote feature (LvFtrGroup::DeviceRemote).

GevSupportedOption Returns if the selected GEV option is supported. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentIPConfigurationLLA Indicates if Link Local Address IP configuration scheme is activated on the given network interface. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentIPConfigurationDHCP Indicates if DHCP IP configuration scheme is activated on the given network interface. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentIPConfigurationPersistentIP Indicates if persistent IP configuration scheme is activated on the given network interface. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentIPAddress Reports the IP address for the given network interface. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentSubnetMask Provides the subnet mask of the given interface. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevCurrentDefaultGateway Indicates the default gateway IP address to be used on the given network interface. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPersistentIPAddress Indicates the persistent IP address for this network interface. It is only used when the device boots with the persistent IP configuration scheme. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPersistentSubnetMask Indicates the persistent subnet mask associated with the persistent IP address on this network interface. It is only used when the device boots with the Persistent IP configuration scheme. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPersistentDefaultGateway Indicates the persistent default gateway for this network interface. It is only used when the device boots with the persistent IP configuration scheme. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevNumberOfInterfaces Indicates the number of physical network interfaces supported by this device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMessageChannelCount Indicates the number of message channels supported by this device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevStreamChannelCount Indicates the number of stream channels supported by this device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevHeartbeatTimeout Indicates the current heartbeat timeout in milliseconds. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevTimestampTickFrequency Indicates the number of timestamp ticks during 1 second (frequency in Hz). LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevTimestampControlLatch Latches current timestamp counter into GevTimestampValue. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

GevTimestampControlReset Resets the Timestamp counter to 0. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

GevTimestampControlLatchReset Reset and latch in a single command. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

GevTimestampValue Returns the latched 64-bit value of the timestamp counter. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevCCP Controls the device access privilege of an application. LvFtrType::Enumeration. Values: LvGevCCP. Device remote feature (LvFtrGroup::DeviceRemote).

GevStreamChannelSelector Selects the stream channel to control. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPInterfaceIndex Index of network interface to use. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPHostPort Indicates the port to which the device must send data stream. Setting this value to 0 closes the stream channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPSFireTestPacket Sends a test packet. When this feature is set, the device will fire one test packet. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPSDoNotFragment The state of this feature is copied into the "do not fragment" bit of IP header of each stream packet. It can be used by the application to prevent IP fragmentation of packets on the stream channel. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPSBigEndian Specifies the stream packet size in bytes to send on this channel. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPSPacketSize Specifies the stream packet size in bytes to send on this channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPD Indicates the delay (in timestamp counter unit, which is currently a microsecond) to insert between each packet for this stream channel. This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCDA Indicates the destination IP address for this stream channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevLinkSpeed Indicates the speed of transmission negotiated by the given network interface in Mb/s. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

UserSetSelector Selects the feature configuration user set to load, save or configure. LvFtrType::Enumeration. Values: LvUserSetSelector. Device remote feature (LvFtrGroup::DeviceRemote).

UserSetLoad Loads the selected user configuration set and makes it active LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

UserSetSave Saves the current device configuration into the selected user configuration set. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

UserSetDefaultSelector Selects the default feature configuration set to be loaded and activated upon camera boot or reset. LvFtrType::Enumeration. Values: LvUserSetDefaultSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkModeActive Activates the chunk mode, ie. inclusion of chunk data in the payload data. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkSelector Selects the chunk to configure. LvFtrType::Enumeration. Values: LvChunkSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkEnable Enables the inclusion of the selected chunk in the payload data. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkOffsetX X offset applied the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkOffsetY Y offset applied the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkWidth Width of the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkHeight Height of the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkPixelFormat Pixel format of the image included in the payload. LvFtrType::Enumeration. Values: see LvPixelFormat. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLinePitch Line pitch of the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkFrameID Frame id of the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkTimestamp Timestamp associated with the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkExposureTime Exposure time used to acquire the image included in the payload. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkGainSelector Selects the gain type to be reported in chunk data. LvFtrType::Enumeration. Values: LvChunkGainSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkGain Gain used to acquire the image included in the payload. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkBlackLevel Black level used to acquire the image included in the payload. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLineStatusAll Bit field indicating staut of all i/o lines at the time the image included in the payload was acquired. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvExternalADCSelector Selects the external ADC to be reported in chunk data. LvFtrType::Enumeration. Values: LvChunkLvExternalADCSelector. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvExternalADCValue Reads the value of the selected external ADC at time of acquisition of the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventSelector Selects which event to signal to the host application. LvFtrType::Enumeration. Values: LvEventSelector. Device remote feature (LvFtrGroup::DeviceRemote).

EventNotification Activate or deactivate the notification to the host application of the selected event occurrence. LvFtrType::Enumeration. Values: LvEventNotification. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppID ID string the smart application LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt1 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt2 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt3 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt4 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt5 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt6 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt7 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt8 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt9 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt10 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt11 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt12 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt13 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt14 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt15 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt16 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt17 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt18 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt19 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt20 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt21 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt22 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt23 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt24 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt25 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt26 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt27 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt28 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt29 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt30 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt31 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppInt32 Generic signed integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint1 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint2 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint3 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint4 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint5 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint6 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint7 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint8 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint9 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint10 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint11 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint12 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint13 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint14 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint15 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint16 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint17 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint18 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint19 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint20 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint21 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint22 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint23 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint24 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint25 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint26 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint27 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint28 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint29 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint30 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint31 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppUint32 Generic unsigned integer register controlling a smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppAsciiCmdString Characters of the ASCII command for the smart application. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppAsciiCmdExecute Executes the ASCII command for the smart application. LvFtrType::↩ Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppAsciiCmdFeedback Response to the ASCII command for the smart application. LvFtrType::↩ String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppAsciiCmdRetCode Numeric return code of the ASCII command for the smart application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppPath Path of the smart application to be started LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppStart Starts the smart application defined by the path. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvLog Returns the unique identifier of the log type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvLogTimestamp Returns the timestamp of the log event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvLogMessage The log message coming with the event LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppLog Returns the unique identifier of the smart application log type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppLogTimestamp Returns the timestamp of the smart application Smart Application Log Event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppLogMessage The smart application Smart Application Log message coming with the event. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortBaudRate Baud rate used for the serial port communication. LvFtrType::Enumeration. Values: LvSerialPortBaudRate. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortParity Parity used for the serial port communication. LvFtrType::Enumeration. Values: LvSerialPortParity. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortDataBits Data bits per character for the serial port communication. LvFtrType::Enumeration. Values: LvSerialPortDataBits. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortStopBits Stop bits per character for the serial port communication. LvFtrType::Enumeration. Values: LvSerialPortStopBits. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortTimeout Timeout value used to finish waiting for command response LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortEOTMarker Short string (or single character) marking end of transmission. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortMaxResponseLength Maximal expected length of the command response. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortCommandString String of the ASCII command to be sent over the serial port. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortCommandSend Sends the ASCII command over the serial port Command. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortCommandResponse Response to the ASCII command sent over the serial port StringReg. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

LvSerialPortCommandStatus Status code indicating success of the last command. Values: LvSerialPortCommandStatus. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

LvSmartAppExitEvent Sends an exit event to the running smart application. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

LvWatchdogTimerValue Current watchdog timer value - reports the current value, after which the timer expires and the device reboots. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlInvertedPolarity Indicates if the lens is wired with inverted polarity, meaning that driving the power switch to the plus side decreases the external ADC feedback. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevMCPHostPort Controls the port to which the device must send messages. Setting this value to 0 closes the message channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMCDA Controls the destination IP address for the message channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMCTT Provides the transmission timeout value in milliseconds. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMCRC Controls the number of retransmissions allowed when a message channel message times out. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppString The smart application string related to the delivered payload. LvFtrType::String Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppIntSelector Selects one of the signed integer values related to the delivered payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppInt The selected smart application signed integer related to the delivered payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppUintSelector Selects one of the unsigned integer values related to the delivered payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppUint The selected smart application unsigned integer related to the delivered payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvSmartAppRegister The smart application raw register related to the delivered payload. LvFtrType::Buffer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppString Returns the unique identifier of the smart application string type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppStringTimestamp Returns the timestamp of the smart application string event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppStringValue The smart application string value coming with the event. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppInt Returns the unique identifier of the smart application signed integer type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppIntTimestamp Returns the timestamp of the smart application signed integer event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppIntSelector Selects one of the signed integer values coming with the event. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppIntValue Value of the selected signed integer coming with the event. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppUint Returns the unique identifier of the smart application unsigned integer type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppUintTimestamp Returns the timestamp of the smart application unsigned integer event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppUintSelector Selects one of the unsigned integer values coming with the event. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppUintValue Value of the selected unsigned integer coming with the event. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppRegister Returns the unique identifier of the smart application raw register type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppRegisterTimestamp Returns the timestamp of the smart application raw register event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvSmartAppRegisterValue The smart application raw register value coming with the event. LvFtrType::String. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceSFNCVersionMajor Major version of the Standard Feature Naming Convention that was used to create the device's XML. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceSFNCVersionMinor Minor version of the Standard Feature Naming Convention that was used to create the device's XML. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceSFNCVersionSubMinor Sub-minor version of Standard Feature Naming Convention that was used to create the device's XML. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLineDebounceDuration Sets the duration (in microseconds) of the line debounce period. Value of 0.0 switches the debouncer off. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

ActionDeviceKey Provides the device key that allows the device to check the validity of action commands. The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ActionSelector Selects to which action signal further action settings apply. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ActionGroupKey Provides the key that the device will use to validate the action on reception of the action protocol message. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ActionGroupMask Provides the mask that the device will use to validate the action on reception of the action protocol message. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvLensControlCalibrationStatus Reports current calibration status of the selected power switch and its bound ADC. The status is computed from the current ADC range, no matter if it is a result of calibration operation or configured manually. LvFtrType::Enumeration. Values: LvLensControlCalibrationStatus. Device remote feature (LvFtrGroup::DeviceRemote).

LvLUTMode Selects the LUT control mode. LvFtrType::Enumeration. Values: LvLUTMode. Device remote feature (LvFtrGroup::DeviceRemote).

BalanceRatioSelector Selects which color channel to configure for white-balancing. LvFtrType::Enumeration. Values: LvBalanceRatioSelector. Device remote feature (LvFtrGroup::DeviceRemote).

BalanceRatio Controls white balance ratio coefficient to be applied on the selected color channel. Note that the white balance functionality is implemented using the LUT. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

BalanceWhiteAuto Controls the mode for automatic white balancing between the color channels. The white balancing ratios are automatically adjusted. Note that the white balance functionality is implemented using the LUT. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

GevDeviceClass Returns the class of the device. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

GevIPConfigurationStatus Reports the current IP configuration status, ie. the method through which the network interface was configured. LvFtrType::Enumeration. Values: LvGevIPConfigurationStatus. Device remote feature (LvFtrGroup::DeviceRemote).

GevDiscoveryAckDelay Indicates the maximum randomized delay the device will wait to acknowledge a discovery command. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevGVCPExtendedStatusCodes Enables the generation of extended status codes. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevGVCPPendingAck Enables the generation of PENDING_ACK. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevGVCPHeartbeatDisable Disables the GVCP heartbeat. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevGVCPPendingTimeout Indicates the longest GVCP command execution time before a device returns a PENDING_ACK. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPrimaryApplicationSwitchoverKey Controls the key to use to authenticate primary application switchover requests. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPrimaryApplicationSocket Returns the UDP source port of the primary application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevPrimaryApplicationIPAddress Returns the address of the primary application. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevMCSP This feature indicates the source port for the message channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCCFGUnconditionalStreaming Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages). LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCCFGExtendedChunkData Enables cameras to use the extended chunk data payload type for this stream channel. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

GevSCPDirection Reports the direction of the stream channel. LvFtrType::Enumeration. Values: LvGevSCPDirection Device remote feature (LvFtrGroup::DeviceRemote).

GevSCSP Indicates the source port of the stream channel. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvTriggerDelayed Flag indicating if the trigger was delayed when acquiring the image included in the payload. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvTriggerDropped Returns the unique identifier of the dropped trigger type of event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

EventLvTriggerDroppedTimestamp Returns the timestamp of the dropped trigger event. It can be used to determine precisely when the event occured. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeEnable Selects the LED clusters of the strobe light that should be enabled. LvFtrType::Enumeration. Values: LvStrobeEnable. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeDurationMode Controls the way how the maximum time of strobe duration is calculated. LvFtrType::Enumeration. Values: LvStrobeDurationMode. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeDuration Duration of the strobe pulse in usec. The maximum time depends on the setting of Strobe Duration Mode feature. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeDelay A delay before the strobe pulse starts after frame trigger is applied in usec. LvFtrType::Float. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeBrightness Brighness (in %) of the strobe light. Allows to lower the full brighness of the strobe. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvStrobeDropMode Sets mode of handling of strobes not matching the device hardware constraints. If a strobe is required (activated by a frame trigger) before the strobe device is ready, the strobe must be dropped or delayed. LvFtrType::Enumeration. Values: LvStrobeDropMode. Device remote feature (LvFtrGroup::DeviceRemote).

LvLUTReset Resets the LUT settings. LvFtrType::Command. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvStrobeDropped Flag indicating if the configured strobe was dropped when acquiring the image included in the payload. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ReverseX Flip horizontally the image sent by the device. The AOI is applied after the flipping. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ReverseY Flip horizontally the image sent by the device. The AOI is applied after the flipping. LvFtrType::↩ Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

RegionSelector Selects the region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

RegionMode Controls if the selected Region of interest is active and streaming. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

RegionDestination Control the destination of the selected region. LvFtrType::Enumeration. Device remote feature (LvFtrGroup::DeviceRemote).

AcquisitionFrameCount Control the number of frames to acquire in MultiFrame Acquisition mode. LvFtr↩ Type::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

AcquisitionBurstFrameCount Control the number of frames to acquire for each FrameBurstStart trigger. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvCustomID Revision number of the custom module. LvFtrType::String. Device remote feature (LvFtrGroup↩ ::DeviceRemote).

LvCustomInfo Info register in the custom module. LvFtrType::Integer. Device remote feature (LvFtrGroup::↩ DeviceRemote).

LvCustomRegMode Controls the way of addressing a register in the custom module. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvCustomRegAddr Defines the address of a register in the custom module. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvCustomRegData Transfers data to and from a register in the custom module. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LvCustomRegMux Defines the address and transfers data to and from a register in the custom module. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

LinePitch Total number of bytes between 2 successive lines. This feature is used to facilitate alignment of image data. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvFrameAbort Flag indicating if a framee was dropped when acquiring the image included in the payload. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvTriggerDropped Flag indicating if a trigger was dropped when acquiring the image included in the payload. LvFtrType::Boolean. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvTriggerError Flag indicating if a missmatch between trigger and sensor data was detected when acquiring the image included in the payload. LvFtrType::Boolean. Device remote feature (LvFtrGroup::↩ DeviceRemote).

ChunkLvEncoderPosition Encoder position generating the trigger for the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

ChunkLvEncoderRotation Encoder rotation generating the trigger for the image included in the payload. LvFtrType::Integer. Device remote feature (LvFtrGroup::DeviceRemote).

DeviceID Device ID. LvFtrType::String. Device GenTL feature (LvFtrGroup::DeviceGtl).

DeviceType Device type. LvFtrType::Enumeration. Values: LvDeviceType. Device GenTL feature (LvFtr↩ Group::DeviceGtl).

GevDeviceIPAddress Device IP address. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::Device↩ Gtl).

GevDeviceSubnetMask Device subnet mask. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::↩ DeviceGtl).

GevDeviceMACAddress Device MAC address. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::↩ DeviceGtl).

GevDeviceGateway Device gateway. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevDeviceStreamCaptureMode Device stream capture mode. Controls, how the GVSP (image) stream is processed by the host machinge. LvFtrType::Enumeration. Values: LvGevDeviceStreamCaptureMode. Device GenTL feature (LvFtrGroup::DeviceGtl).

StreamSelector Stream selector. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::DeviceGtl).

StreamID Stream ID. LvFtrType::String. Device GenTL feature (LvFtrGroup::DeviceGtl).

DeviceEndianessMechanism Identifies the endianess mode. This feature should be set to 'standard' for all GigE Vision remote devices based on GenICam schema version 1.1 (and newer). The value 'legacy' is intended for devices using GenICam schema version 1.0. Values: LvDeviceEndianessMechanism. LvFtrType::Enumeration. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevFindMaxPacketSize Determine the maximal usable packet size for streaming. The procedure is already applied automatically when opening the device. Do not use during active streaming. LvFtrType::Command. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevPacketSizeValue Streaming packet size to be verified using the Test Packet Size command. Do not use during active streaming. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevTestPacketSize Test if the selected packet size is suitable for streaming in current network configuration. LvFtrType::Command. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevPacketSizeTestSuccess Reports success of the last packet size test command. LvFtrType::Boolean. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevCCTT Controls the GigE Vision control channel transmission timeout value in milliseconds. When it expires, the acknowledge from the device will be considered as missing and the command eventually sent again. LvFtrType::Command. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvGevCCRC Controls the number of GigE Vision control channel command retransmissions allowed when a control channel command times out. LvFtrType::Command. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvCCStatus Control channel status. LvFtrType::Integer. Device GenTL feature (LvFtrGroup::DeviceGtl).

LvDeviceDisplayName User readable name of the device. LvFtrType::String. Device local feature (LvFtrGroup::DeviceLocal).

LvDeviceIsAcquiring Returns true if the acquisition was started. Note that this feature does not tell whether the images are actually delivered to the output buffer queue; it simply informs that your application is between the AcquisitionStart and AcquisitionStop actions. LvFtrType::Boolean. Device local feature (LvFtrGroup::DeviceLocal).

LvUniProcessMode The UniProcessing provides unified API for image preprocessing, which is done either on the device itself, if it is supported by the hardware, or by software, if not. The preprocessing includes Bayer decoding or pixel format conversion, application of LUT and Color Correction. LvFtrType::Enumeration. Values: LvUniProcessMode. Device local feature (LvFtrGroup::DeviceLocal).

LvUniProcessEnableInPlace If possible, the software image preprocessing will be preferably done in the same image (not to another buffer). This is possible only in case the preprocessing does not change the pixel format, that means the LvUniPixelFormat must be equal to PixelFormat (for example the Bayer decoding is not done by software). LvFtrType::Boolean. Device local feature (LvFtrGroup::DeviceLocal).

LvUniPixelFormat If the image preprocessing is enabled, this is the desired pixel format, to which the image is to be converted. Only monochrome and RGB/BGR color pixel formats are supported. The processing chain is set so that:

- if the PixelFormat is undecoded Bayer, the Bayer decoding to desired LvUniPixelFormat is included
- otherwise if the PixelFormat is not equal to LvUniPixelFormat, a pixel format conversion is included.

LvFtrType::Enumeration. Values: see LvPixelFormat. Device local feature (LvFtrGroup::DeviceLocal).

LvUniProcessPayloadSize Returns the size needed for the processing output buffer, which is to be used when the in-place processing is not enabled or not possible. Normally is this buffer allocated automatically for each acquisition buffer, but your application can also provide your own buffers and this feature gives the size of the buffers needed. LvFtrType::Integer. Device local feature (LvFtrGroup::DeviceLocal).

LvUniLinePitch The line increment of the process buffer, if the processing is active, or of the source buffer, if processing is not active. To access the image regardless whether it was processed to the process buffer or not, you need 5 independent values:

- Pointer to the data - use LvUniBase feature of the Buffer, which points either to the acquired image (if no processing was done), or to the processed image (if it was processed).

- Width and height - these are the same for the acquired and processed image, so use the Width and Height from the remote device, or ChunkWidth and ChunkHeight if these can change during the acquisition.
- Pixel format - use LvUniPixelFormat - if this is different from the PixelFormat then processing is done, so LvUniPixelFormat is always correct.
- Line pitch - use LvUniLinePitch, which returns proper line pitch of the buffer, to which the LvUniBase pointer points. Note that the above is valid only in case the processing can be successfully done (for example the source image is not in unsupported PixelFormat) and is not disabled (for example by LvUniProcessExecution=OnExplicitRequest). LvFtrType::Integer. Device local feature (LvFtrGroup↩ ::DeviceLocal).

LvUniBayerDecoderAlgorithm Selects the Bayer array decoding method for the software processing. This does not apply to the hardware Bayer decoding on the device, which is usually fixed to one method. LvFtrType::Enumeration. Values: LvBayerDecoderAlgorithm. Device local feature (LvFtrGroup::Device↩ Local).

LvUniBrightness Brightness of the image. It is realized by the LUT. Values under 1.0 means darker than original, above 1.0 lighter than the original. The LvUniLUTMode must be Generated, in order to enable this feature. LvFtrType::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniContrast Contrast of the image. It is realized by the LUT. Values under 1.0 means lower contrast than original, above 1.0 higher contrast than the original. The LvUniLUTMode must be Generated, in order to enable this feature. LvFtrType::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniGamma Gamma correction of the image. It is realized by the LUT. Values under 1.0 make the middle tones darker, above 1.0 lighter. The LvUniLUTMode must be Generated, in order to enable this feature. LvFtrType::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniBalanceRatioSelector Selects which color channel will be accessed by the LvUniBalanceRatio feature. The LvUniLUTMode must be Generated, in order to enable this feature. LvFtrType::Enumeration. Values: LvUniBalanceRatioSelector. Device local feature (LvFtrGroup::DeviceLocal).

LvUniBalanceRatio The white balance factor to be applied on the selected color channel. The selected color channel of all pixels will be multiplied by this value (not directly, but through the precalucated LUT). If the value is $<$ 1.0, the saturated pixels will become gray (white is no more white). Thus it is better if all 3 factors are greater than or equal to 1.0. LvFtrType::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniBalanceWhiteAuto Selects the action for automatic white balance calculation. Currently only the option Once is available. Setting this option causes the following:

- if there is already acquired image available, the white balance factors are calculated from this image and LUT is updated to reflect the changes
- if there is no image acquired yet, an internal flag is set and the calculation is done when the image is acquired. Note that the enumeration is self-clearing, that means its value is automatically changed to Off, when the white balance calculation is finished. The newly calculated white balance is applied to to newly acquired images, not to the existing ones, unless you explicitly call the ExecProcess command for the already acquired buffers. At the time of calculation the camera should look at a neutral grey (not white) object, which should fill the whole image area. Making white balance from normal image can bring less satisfactory results. LvFtrType::Enumeration. Values: LvUniBalance↩ WhiteAuto. Device local feature (LvFtrGroup::DeviceLocal).

LvUniBalanceWhiteReset Sets all the white balance factors (LvUniBalanceRatio) to 1. The advantage of this feature in comparison with setting the 3 factors to 1 is that the LUT is updated only once, so it is faster. LvFtrType::Command. Device local feature (LvFtrGroup::DeviceLocal).

LvUniColorTransformationSelector Selects which color transformation module is controlled by the color transformation features. It also gives particular meaning to individual color transformation gains. Lv↩ FtrType::Enumeration. Values: LvUniColorTransformationSelector. Device local feature (LvFtrGroup::↩ DeviceLocal).

LvUniColorTransformationEnable Enables the Color Transformation in the processing. When disabled, the Color Transformation matrix does not lose its values; when enabling it, the original values are retained. LvFtrType::Boolean. Device local feature (LvFtrGroup::DeviceLocal).

LvUniColorTransformationValueSelector Selects the cell of the Color Transformation matrix to be accessed by LvUniColorTransformationValue. LvFtrType::Enumeration. Values: LvColorTransformationValue↩Selector. Device local feature (LvFtrGroup::DeviceLocal).

LvUniColorTransformationValue The value of the selected cell of the Color Transformation matrix. LvFtr↩Type::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniSaturation Sets the Color Correction matrix according to specified saturation. The saturation set to 0 causes a conversion to greyscale, 1.0 leaves the image identical, 2.0 emphasizes the colors. LvFtrType↩::Float. Device local feature (LvFtrGroup::DeviceLocal).

LvUniProcessExecution Defines the point, when the software image processing of the buffer is done. You may need to define this point in case you do not need all the images to be processed. Note that this applies only to the software processing; the hardware processing is done on the remote device always. LvFtrType::Enumeration. Values: LvUniProcessExecution. Device local feature (LvFtrGroup::Device↩Local).

LvUniLUTMode Selects the LUT control mode. The mode determines, if the LUT can be directly modified by the application, or if the LUT is to be reserved for implementation of white balance, gamma, brightnesss and contrast - in such case the LUT is filled with precalculated values by SynView library and cannot be directly modified. LvFtrType::Enumeration Values: LvUniLUTMode. Device local feature (LvFtrGroup::↩DeviceLocal).

LvUniLUTSelector This selector selects for which LUT is applied LvUniLUTIndex/LvUniLUTValue. In case of monochrome image the LUT has only one array = Luminance. In case of color images, the LUT consists of 3 arrays, for Red, Green and Blue. LvFtrType::Enumeration Values: LvUniLUTSelector. Device local feature (LvFtrGroup::DeviceLocal).

LvUniLUTEnable Enables the LUT in the processing. When disabled, the LUT does not lose its values, the disabled LUT is substitued by a linear LUT, and when enabling the LUT, the original values are retained. LvFtrType::Boolean Device local feature (LvFtrGroup::DeviceLocal).

LvUniLUTIndex Index of the element to be accessed in the selected LUT via the LvUniLUTValue feature. Note that accessing the whole LUT by this approach can be very time consuming, namely on Gig↩E cameras. If possible, it is better to use the LvUniLUTValueAll or SynView dedicated LUT functions. LvFtrType::Integer Device local feature (LvFtrGroup::DeviceLocal).

LvUniLUTValue Value of the element for the current LvUniLUTIndex in the selected LUT. Note that accessing the whole LUT by this approach can be very time consuming, namely on GigE cameras. If possible, it is better to use the LvUniLUTValueAll or SynView dedicated LUT functions. LvFtrType::Integer Device local feature (LvFtrGroup::DeviceLocal).

LvUniLUTValueAll This feature enables to get/set the entire content of the selected LUT in one block access. Beware that the LUT buffer structure is vendor and model dependent, so take care if your application is exptected to work with various types od devices or devices from various vendors. LvFtrType::Buffer Device local feature (LvFtrGroup::DeviceLocal).

LvUniColorTransformationMode Selects the Color Transformation matrix control mode. The mode determines, if the matrix can be directly modified by the application, or if the matrix is to be reserved for implementation of the Saturation or other higher level features - in such case the matrix is filled with precalculated values by SynView library and cannot be directly modified. LvFtrType::Enumeration Values: LvUniColorTransformationMode. Device local feature (LvFtrGroup::DeviceLocal).

Info Constant to be used in LvGetInfo() and LvGetInfoStr() to obtain various info about the device.

6.5.2.3 enum LvEventFtr : LvFeature [strong]

LvEventFtr constants.

Enumerator

EventType Represents the GenTL EVENT_EVENT_TYPE info - The event type. LvFtrType::Integer.

NumInQueue Represents the GenTL EVENT_NUM_IN_QUEUE info - Number of events in the event data queue. LvFtrType::Integer.

NumFired Represents the GenTL EVENT_NUM_FIRED info - Number of events, that were fired since the creation of the event. LvFtrType::Integer

6.5.2.4 enum LvInterfaceFtr : LvFeature [strong]

LvInterfaceFtr constants.

Enumerator

InterfaceID Interface ID. LvFtrType::String. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

InterfaceType Interface type. LvFtrType::Enumeration. Values: LvInterfaceType. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceGatewaySelector Interface gateway selector. LvFtrType::Integer. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceGateway Interface gateway. LvFtrType::Integer. Depends on LvInterface::GevInterfaceGatewaySelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevMACAddress Interface MAC address. LvFtrType::Integer. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceSubnetSelector Interface subnet selector. LvFtrType::Integer. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceSubnetIPAddress Interface subnet IP address. LvFtrType::Integer. Depends on LvInterface::GevInterfaceSubnetSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceSubnetMask Interface subnet mask. LvFtrType::Integer. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceUpdateList Update internal list ofdevices. LvFtrType::Command. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceSelector Device selector. LvFtrType::Integer. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceID Device ID. LvFtrType::String. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceVendorName Device vendor name. LvFtrType::String. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceModelName Device Model name. LvFtrType::String. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

DeviceAccessStatus Device access status. LvFtrType::Enumeration. Values: LvDeviceAccessStatus. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevDeviceIPAddress Device IP address. LvFtrType::Integer. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevDeviceSubnetMask Device subnet mask. LvFtrType::Integer. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

GevDeviceMACAddress Device MAC address. LvFtrType::Integer. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

LvInterface_LvDeviceUserID Device User ID. LvFtrType::String. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

LvInterface_LvDeviceSerialNumber Device identifier (serial number). LvFtrType::String. Depends on LvInterface::DeviceSelector. Interface GenTL feature (LvFtrGroup::InterfaceGtl).

LvInterfaceDisplayName User readable name of the interface. LvFtrType::String. Interface local feature (LvFtrGroup::InterfaceLocal).

Info Constant to be used in LvGetInfo() and LvGetInfoStr() to obtain various info about the interface.

6.5.2.5 enum LvRendererFtr : LvFeature [strong]

LvRendererFtr constants.

Enumerator

LvAutoDisplay If set, the image is automatically displayed before it is passed to the supplied callback. This is functional only in case the Event thread is started. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvRenderType Controls way how the acquired images are rendered on the screen. Note that all the Scale-options require scaling capability of the display and might not be supported in all operating systems. LvFtrType::Enumeration. Values: LvRenderType. Renderer local feature (LvFtrGroup::RendererLocal).

LvOffsetX Sets the horizontal offset of the image to be rendered, i.e. the distance from the left edge of the display window. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvOffsetY Sets the vertical offset of the image to be rendered, i.e. the distance from the top edge of the display window. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvWidth Sets the width of the rectangle to which the image is to be rendered. Note that if the LvIgnoreAspectRatio feature is False, the real image width can be smaller, in order to keep the aspect ratio. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvHeight Sets the height of the rectangle to which the image is to be rendered. Note that if the LvIgnoreAspectRatio feature is False, the real image height can be smaller, in order to keep the aspect ratio. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvIgnoreAspectRatio Allows to ignore the original aspect ratio while rendering the image, so the image can be scaled up/down in one dimension with different factor than in the other dimension. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvDisableScaleUp Disables scaling the image up. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvDisableScaleDown Disables scaling the image down. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvCenterImage If the image is smaller than required window client size or the specified rectangle, the image is paced to the center. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvNumberOfTiles Sets the number of tiles used for image rendering. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvColumns Sets the number of columns used for image rendering. Set to 0 if the number of columns is to be calculated automatically. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvRows Sets the number of rows used for image rendering. Set to 0 if the number of rows is to be calculated automatically. LvFtrType::Integer. Renderer local feature (LvFtrGroup::RendererLocal).

LvTileGap Sets the gap between the tiles in pixels. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

LvAutoTileCalculation When set to true, the tile sizes and positions are calculated automatically. When the Display Columns and/or Display Rows are 0, also the number of columns and/or rows is calculated automatically. LvFtrType::Boolean. Renderer local feature (LvFtrGroup::RendererLocal).

Info Constant to be used in LvGetInfo() and LvGetInfoStr() to obtain various info about the device.

6.5.2.6 enum LvStreamFtr : LvFeature [strong]

LvStreamFtr constants.

Enumerator

StreamID Stream ID. LvFtrType::String. Stream GenTL feature (LvFtrGroup::StreamGtl).

StreamAnnouncedBufferCount Number of buffers announced for the data stream. LvFtrType::Integer. Stream GenTL feature (LvFtrGroup::StreamGtl).

StreamAcquisitionModeSelector Selects desired acquisition mode. LvFtrType::Enumeration. Values: LvStreamAcquisitionModeSelector. Stream GenTL feature (LvFtrGroup::StreamGtl).

StreamAnnounceBufferMinimum Minimum number of buffers to be announced for selected acquisition mode. LvFtrType::Integer. Stream GenTL feature (LvFtrGroup::StreamGtl).

StreamType Stream type. LvFtrType::Enumeration. Values: LvStreamType. Stream GenTL feature (LvFtrGroup::StreamGtl).

LvStreamDisplayName Returns the display name of the stream. LvFtrType::String. Stream local feature (LvFtrGroup::StreamLocal).

LvCalcPayloadSize Returns the payload size (size of buffer to hold the image data). If the payload size is not provided by the stream or device, it is calculated, so this feature returns always a valid value. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvPostponeQueueBuffers Number of buffers to be kept postponed before returning to the input buffer pool. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvAwaitDeliveryLimit Limit for images in the output buffer. Applicable only if the event thread is running - then if there is more than this number of buffers in the output queue, the oldest buffers are discarded and returned to input buffer pool. This is useful in case the application is not able to process all the images in time. LvFtrType::Integer. If the value is 0, this limit is inactive. Stream local feature (LvFtrGroup::StreamLocal).

LvAutoAllocateProcessBuffers Enable the auto allocation of process buffers. The process buffers are allocated only if they are needed for the image processing or conversion. You can disable the automatic buffer alocation and provide own buffers, using the LvBuffer::AttachProcessBuffer() function. LvFtrType::Boolean. Stream local feature (LvFtrGroup::StreamLocal).

LvPreallocateProcessBuffers Preallocates all the process buffers, even if it is not yet sure if they will be needed. With this command you can avoid time delays when allocating the buffers during the acquisition. LvFtrType::Command. Stream local feature (LvFtrGroup::StreamLocal).

LvNumDelivered Number of acquired frames since last acquisition start. It is equivalent to the GenTL STREAM_INFO_NUM_DELIVERED info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvNumUnderrun Number of lost frames due to input buffer pool underrun since stream open. It is equivalent to the GenTL STREAM_INFO_NUM_UNDERRUN info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvNumAnnounced Number of announced buffers. It is equivalent to the GenTL STREAM_INFO_NUM_ANNOUNCED info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvNumQueued Number of buffers currently in the input pool. It is equivalent to the GenTL STREAM_INFO_NUM_QUEUED info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvNumAwaitDelivery Number of buffers currently in the output queue. It is equivalent to the GenTL STREAM_INFO_NUM_AWAIT_DELIVERY info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvIsGrabbing Flag indicating whether the acquisition engine is started or not. This is independent from the acquisition status of the remote device. It is equivalent to the GenTL STREAM_INFO_IS_GRABBING info. LvFtrType::Boolean. Stream local feature (LvFtrGroup::StreamLocal).

LvNumAborted Number of aborted frames since last acquisition start. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

LvNumStarted Number of started frames since stream open. It is equivalent to the GenTL STREAM_INFO_NUM_STARTED info. LvFtrType::Integer. Stream local feature (LvFtrGroup::StreamLocal).

Info Constant to be used in LvGetInfo() and LvGetInfoStr() to obtain various info about the stream.

6.5.2.7 enum LvSystemFtr : LvFeature [strong]

LvSystemFtr constants.

Enumerator

TLVendorName GenTL producer vendor name. LvFtrType::String. System GenTL feature (LvFtrGroup::SystemGtl).

TLModelName GenTL producer model name. LvFtrType::String. System GenTL feature (LvFtrGroup::SystemGtl).

TLID GenTL producer ID. LvFtrType::String. System GenTL feature (LvFtrGroup::SystemGtl).

TLVersion GenTL producer version. LvFtrType::String. System GenTL feature (LvFtrGroup::SystemGtl).

TLPath GenTL producer path. LvFtrType::String. System GenTL feature (LvFtrGroup::SystemGtl).

TLType GenTL producer type. LvFtrType::Enumeration. Values: LvTLType. System GenTL feature (LvFtrGroup::SystemGtl).

GenTLVersionMajor GenTL version major. LvFtrType::Integer. System GenTL feature (LvFtrGroup::SystemGtl).

GenTLVersionMinor GenTL version minor. LvFtrType::Integer. System GenTL feature (LvFtrGroup::SystemGtl).

GevVersionMajor GigE Vision version major. LvFtrType::Integer. System GenTL feature (LvFtrGroup::SystemGtl).

GevVersionMinor GigE Vision version minor. LvFtrType::Integer. System GenTL feature (LvFtrGroup::SystemGtl).

InterfaceUpdateList Update internal list of interfaces. LvFtrType::Command. System GenTL feature (LvFtrGroup::SystemGtl).

InterfaceSelector Interface selector. LvFtrType::Integer. System GenTL feature (LvFtrGroup::SystemGtl).

InterfaceID Interface ID. LvFtrType::String. Depends on LvSystem::InterfaceSelector. System GenTL feature (LvFtrGroup::SystemGtl).

GevInterfaceMACAddress Interface MAC address. LvFtrType::Integer. Depends on LvSystem::InterfaceSelector. System GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceDefaultIPAddress Interface default IP address. LvFtrType::Integer. Depends on LvSystem::InterfaceSelector. System GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceDefaultSubnetMask Interface default subnet mask. LvFtrType::Integer. Depends on LvSystem::InterfaceSelector. System GenTL feature (LvFtrGroup::InterfaceGtl).

GevInterfaceDefaultGateway Interface default gateway. LvFtrType::Integer. Depends on LvSystem::InterfaceSelector. System GenTL feature (LvFtrGroup::InterfaceGtl).

LvSystemDisplayName User readable name of the system. LvFtrType::String. System local feature (LvFtrGroup::SystemLocal).

Info Constant to be used in LvGetInfo() and LvGetInfoStr() to obtain various info about the system.

6.6 SynView .Net Class Library Enumeration Entries

Enumerations

- enum LvPixelFormat : LvEnum {
  Mono8 = 0x01080001, Mono8S = 0x01080002, Mono10 = 0x01100003, Mono10Packed = 0x010C0004,
  Mono12 = 0x01100005, Mono12Packed = 0x010C0006, Mono14 = 0x01100025, Mono16 = 0x01100007,
  BayerGR8 = 0x01080008, BayerRG8 = 0x01080009, BayerGB8 = 0x0108000A, BayerBG8 = 0x0108000B,
  BayerGR10 = 0x0110000C, BayerRG10 = 0x0110000D, BayerGB10 = 0x0110000E, BayerBG10 = 0x0110000F,
  BayerGR12 = 0x01100010, BayerRG12 = 0x01100011, BayerGB12 = 0x01100012, BayerBG12 = 0x01100013,
  BayerGR10Packed = 0x010C0026, BayerRG10Packed = 0x010C0027, BayerGB10Packed = 0x010C0028,
  BayerBG10Packed = 0x010C0029,
  BayerGR12Packed = 0x010C002A, BayerRG12Packed = 0x010C002B, BayerGB12Packed = 0x010C002C,
  BayerBG12Packed = 0x010C002D,
  BayerGR16 = 0x0110002E, BayerRG16 = 0x0110002F, BayerGB16 = 0x01100030, BayerBG16 = 0x01100031,
  RGB8 = 0x02180014, BGR8 = 0x02180015, RGBA8 = 0x02200016, BGRA8 = 0x02200017,
  RGB10 = 0x02300018, BGR10 = 0x02300019, RGB12 = 0x0230001A, BGR12 = 0x0230001B,
  RGB16 = 0x02300033, RGB10V1Packed = 0x0220001C, RGB10P32 = 0x0220001D, RGB12V1Packed = 0x02240034,
  RGB565P = 0x02100035, BGR565P = 0x02100036, YUV411_8 = 0x020C001E, YUV422_8_UYVY = 0x0210001F,
  YUV422_8 = 0x02100032, YUV8 = 0x02180020, YCbCr422_8 = 0x0210003B, YCbCr601_422_8 = 0x0210003E,
  YCbCr601_422_8_CbYCrY = 0x02100044, RGB8_Planar = 0x02180021, RGB10_Planar = 0x02300022,
  RGB12_Planar = 0x02300023,
  RGB16_Planar = 0x02300024, BGR555P = 0x021000E1, Mono8Signed, RGB8Packed,
  BGR8Packed, RGBA8Packed, BGRA8Packed, RGB10Packed,
  BGR10Packed, RGB12Packed, BGR12Packed, RGB16Packed,
  RGB10V2Packed, RGB565Packed, BGR565Packed, YUV411Packed,
  YUV422Packed, YUV422YUYVPacked, YUV444Packed, RGB8Planar,
  RGB10Planar, RGB12Planar, RGB16Planar, Mono8s,
  RGBa8, BGRa8, RGB565p, BGR565p,
  RGB10p32, BGR555p, YUV411_8_UYYVYY, YUV8_UYV }
- enum LvDeviceAccess : LvEnum { None = 1, ReadOnly = 2, Control = 3, Exclusive = 4 }
- enum LvDeviceAccessStatus : LvEnum { Unknown = 0, ReadWrite = 1, ReadOnly = 2, NoAccess = 3 }
- enum LvDeviceScanType : LvEnum { Areascan, Linescan }
- enum LvDeviceClockSelector : LvEnum { SensorDigitization }
- enum LvDeviceTemperatureSelector : LvEnum { Sensor, Mainboard }
- enum LvAOIMode : LvEnum { Automatic, ClipOnTransfer, Manual }
- enum LvAcquisitionMode : LvEnum { SingleFrame, MultiFrame, Continuous }
- enum LvExposureAuto : LvEnum { Off, Once, Continuous }
- enum LvTriggerSelector : LvEnum { FrameStart, FrameBurstStart }
- enum LvTriggerMode : LvEnum { Off, On }
- enum LvTriggerSource : LvEnum {
  Line1, Line2, Line3, Line4,
  Line5, Line6, Line7, Line8,
  Line17, Line18, Line19, Line20,
  Line21, Line22, Line23, Line24,
  Software, Action1, Action2, Action3,
  Action4, Action5, Action6, Action7,
  Action8, Quad, Counter1, Counter2,
  Counter3, Counter4, Timer1, Timer2,
  Timer3, Timer4 }
- enum LvTriggerActivation : LvEnum { RisingEdge, FallingEdge }

- enum LvTriggerCaching : LvEnum { Cache, Drop }
- enum LvExposureMode : LvEnum { Timed }
- enum LvAcquisitionFrameRateControlMode : LvEnum { Off, On }
- enum LvLineSelector : LvEnum {
  Line1, Line2, Line3, Line4,
  Line5, Line6, Line7, Line8,
  Line9, Line10, Line11, Line12,
  Line13, Line14, Line15, Line16,
  Line17, Line18, Line19, Line20,
  Line21, Line22, Line23, Line24,
  Line25, Line26, Line27, Line28,
  Line29, Line30, Line31, Line32 }
- enum LvLineMode : LvEnum { Input, Output }
- enum LvLineFormat : LvEnum {
  NoConnect, TriState, TTL, LVDS,
  RS422, OptoCoupled }
- enum LvLineSource : LvEnum {
  Off, ExposureActive, Timer1Active, Timer2Active,
  Timer3Active, Timer4Active, UserOutput1, UserOutput2,
  UserOutput3, UserOutput4, UserOutput5, UserOutput6,
  UserOutput7, UserOutput8, Counter1, Counter2,
  Counter3, Counter4 }
- enum LvCounterSelector : LvEnum { Counter1, Counter2, Counter3, Counter4 }
- enum LvCounterMode : LvEnum { Autoreset }
- enum LvCounterEventSource : LvEnum {
  Off, FrameTrigger, TimerTick, Line1,
  Line2, Line3, Line4, Line17,
  Line18 }
- enum LvTimerSelector : LvEnum { Timer1, Timer2, Timer3, Timer4 }
- enum LvTimerTriggerSource : LvEnum {
  Off, FrameTrigger, Counter1End, Counter2End,
  Counter3End, Counter4End, UserOutput1, UserOutput2,
  UserOutput3, UserOutput4, UserOutput5, UserOutput6,
  UserOutput7, UserOutput8 }
- enum LvSpecialPurposeTriggerSelector : LvEnum { ImageStampsReset }
- enum LvSpecialPurposeTriggerSource : LvEnum {
  Off, Line1, Line2, Line3,
  Line4, Line5, Line6, Line7,
  Line8, Line17, Line18, Line19,
  Line20, Line21, Line22, Line23,
  Line24, Action1, Action2, Action3,
  Action4, Action5, Action6, Action7,
  Action8 }
- enum LvSpecialPurposeTriggerActivation : LvEnum { RisingEdge, FallingEdge }
- enum LvImageStampSelector : LvEnum { Timestamp, FrameID }
- enum LvBootSwitch : LvEnum { PureGEV, Legacy }
- enum LvGainSelector : LvEnum { All, AnalogAll, DigitalAll }
- enum LvGainAuto : LvEnum { Off, Once, Continuous }
- enum LvBlackLevelSelector : LvEnum { All }
- enum LvBlackLevelAuto : LvEnum { Off, Once, Continuous }
- enum LvColorTransformationSelector : LvEnum { RGBtoRGB }
- enum LvColorTransformationValueSelector : LvEnum {
  Gain00, Gain01, Gain02, Gain10,
  Gain11, Gain12, Gain20, Gain21,
  Gain22 }
- enum LvExternalDeviceControlMode : LvEnum { Custom }
- enum LvExternalADCSelector : LvEnum { ExternalADC1, ExternalADC2, ExternalADC3, ExternalADC4 }

- enum LvPowerSwitchCurrentAction : LvEnum {
  Idle, Pulse, Calibrate, AdjustPosition,
  Drive }
- enum LvPowerSwitchSelector : LvEnum { PowerSwitch1, PowerSwitch2, PowerSwitch3, PowerSwitch4 }
- enum LvPowerSwitchDrive : LvEnum { Off, Plus, Minus }
- enum LvPowerSwitchDriveAll : LvEnum { Off, Plus, Minus }
- enum LvPowerSwitchBoundADC : LvEnum {
  None, ExternalADC1, ExternalADC2, ExternalADC3,
  ExternalADC4 }
- enum LvLensControlTargetApproach : LvEnum { Direct, FromPlus, FromMinus }
- enum LvLUTSelector : LvEnum { Luminance, Red, Green, Blue }
- enum LvGevDeviceModeCharacterSet : LvEnum { UTF8 }
- enum LvGevSupportedOptionSelector : LvEnum {
  IPConfigurationLLA, IPConfigurationDHCP, IPConfigurationPersistentIP, CommandsConcatenation,
  WriteMem, PacketResend, Event, EventData,
  PendingAck, Action, PrimaryApplicationSwitchover, ExtendedStatusCodes,
  DiscoveryAckDelayWritable, DiscoveryAckDelay, TestData, ManifestTable,
  CCPApplicationSocket, LinkSpeed, HeartbeatDisable, SerialNumber,
  UserDefinedName, StreamChannelSourceSocket, StreamChannel0ExtendedChunkData, StreamChannel0UnconditionalStreaming,
  StreamChannel0IPReassembly, StreamChannel0BigAndLittleEndian, MessageChannelSourceSocket }
- enum LvGevCCP : LvEnum { OpenAccess, ExclusiveAccess, ControlAccess, ControlAccessSwitchoverActive }
- enum LvUserSetSelector : LvEnum {
  Default, UserSet1, UserSet2, UserSet3,
  UserSet4 }
- enum LvUserSetDefaultSelector : LvEnum {
  Default, UserSet1, UserSet2, UserSet3,
  UserSet4, None }
- enum LvChunkSelector : LvEnum {
  OffsetX, OffsetY, Width, Height,
  PixelFormat, LinePitch, FrameID, Timestamp,
  ExposureTime, Gain, LineStatusAll, BlackLevel,
  LvExternalADCValue, LvSmartAppString, LvSmartAppInt, LvSmartAppUint,
  LvSmartAppRegister, LvTriggerDelayed, LvStrobeDropped, LvFrameAbort,
  LvTriggerDropped, LvTriggerError, LvEncoderPosition, LvEncoderRotation }
- enum LvChunkGainSelector : LvEnum { AnalogAll, DigitalAll }
- enum LvEventSelector : LvEnum {
  LvLog, LvSmartAppLog, LvSmartAppString, LvSmartAppInt,
  LvSmartAppUint, LvSmartAppRegister, LvTriggerDropped }
- enum LvEventNotification : LvEnum { Off, On }
- enum LvTLType : LvEnum { Mixed, Custom, GEV }
- enum LvInterfaceType : LvEnum { Custom, GEV }
- enum LvDeviceType : LvEnum { Custom, GEV }
- enum LvGevDeviceStreamCaptureMode : LvEnum { SystemDefault, Socket, FilterDriver }
- enum LvStreamAcquisitionModeSelector : LvEnum { Default }
- enum LvStreamType : LvEnum { Custom, GEV }
- enum LvUniProcessMode : LvEnum { HwOnly, SwOnly, Auto, Off }
- enum LvBayerDecoderAlgorithm : LvEnum {
  NearestNeighbour, BilinearInterpolation, BilinearColorCorrection, PixelGrouping,
  VariableGradient }
- enum LvUniBalanceRatioSelector : LvEnum { Red, Green, Blue }
- enum LvUniBalanceWhiteAuto : LvEnum { Off, Once }
- enum LvUniColorTransformationSelector : LvEnum { RGBtoRGB }

- enum LvUniColorTransformationValueSelector : LvEnum { Gain00, Gain01, Gain02, Gain10, Gain11, Gain12, Gain20, Gain21, Gain22 }
- enum LvRenderType : LvEnum { FullSize, ScaleToFit, ScaleToSize, ScaleToTiles }
- enum LvSerialPortBaudRate : LvEnum { Baud2400, Baud4800, Baud9600, Baud14400, Baud19200, Baud38400, Baud57600, Baud115200 }
- enum LvSerialPortParity : LvEnum { None, Odd, Even }
- enum LvSerialPortDataBits : LvEnum { DataBits7, DataBits8 }
- enum LvSerialPortStopBits : LvEnum { StopBits1, StopBits1dot5, StopBits2 }
- enum LvSerialPortCommandStatus : LvEnum { Success, Timeout, PortBusy, CommunicationError, FrameError, ParityError, Overflow }
- enum LvChunkLvExternalADCSelector : LvEnum { ExternalADC1, ExternalADC2, ExternalADC3, ExternalADC4 }
- enum LvUserOutputSelector : LvEnum { UserOutput1, UserOutput2, UserOutput3, UserOutput4, UserOutput5, UserOutput6, UserOutput7, UserOutput8 }
- enum LvUniProcessExecution : LvEnum { OnBufferPtrQuery, OnPopFromQueue, OnExplicitRequest }
- enum LvLensControlCalibrationStatus : LvEnum { Invalid, Valid }
- enum LvLUTMode : LvEnum { Direct, BalanceWhite }
- enum LvBalanceRatioSelector : LvEnum { Red, Green, Blue }
- enum LvBalanceWhiteAuto : LvEnum { Off, Once, Continuous }
- enum LvGevDeviceClass : LvEnum { Transmitter }
- enum LvGevIPConfigurationStatus : LvEnum { None, PersistentIP, DHCP, LLA, ForceIP }
- enum LvGevSCPDirection : LvEnum { Transmitter }
- enum LvDeviceEndianessMechanism : LvEnum { Legacy, Standard }
- enum LvUniLUTMode : LvEnum { Direct, Generated }
- enum LvUniLUTSelector : LvEnum { Luminance, Red, Green, Blue }
- enum LvUniColorTransformationMode : LvEnum { Direct, Generated }
- enum LvStrobeEnable : LvEnum { Off, AllClusters, LEDCluster1, LEDCluster2 }
- enum LvStrobeDurationMode : LvEnum { FrameRateRelated, Free }
- enum LvStrobeDropMode : LvEnum { DropStrobe, DelayFrame }
- enum LvRegionSelector : LvEnum { Region0, Region1, Region2, Region3 }

6.6.1 Detailed Description

6.6.2 Enumeration Type Documentation

6.6.2.1 enum LvAcquisitionFrameRateControlMode : LvEnum [strong]

Enum values for the LvDevice::LvAcquisitionFrameRateControlMode feature.

Enumerator

Off Disables frame rate control - the camera operates at maximum frame rate

On Enables frame rate control - the rate can be explicitly adjusted

6.6.2.2 enum LvAcquisitionMode : LvEnum [strong]

Enum values for the LvDevice::AcquisitionMode feature.

Enumerator

SingleFrame Single frame acquisition - after acquisition starts, single frame is acquired and acquisition stops.

MultiFrame Multiple frame acquisition - after acquisition starts, specified numbr of frames is acquired and acquisition stops.

Continuous Continuous acquisition - after starting, the acquisition is active until explicitly stopped.

6.6.2.3 enum LvAOIMode : LvEnum [strong]

Enum values for the LvDevice::LvAOIMode feature.

Enumerator

Automatic Camera automatically applies as much of the desired AOI setting on the sensor and the rest is cut on transfer

ClipOnTransfer The AOI is applied before the transfer, in camera memory

Manual Fine control of separate AOI setting on the sensor and before the transfer

6.6.2.4 enum LvBalanceRatioSelector : LvEnum [strong]

Enum values for the LvDevice::BalanceRatioSelector feature.

Enumerator

Red Balance ratio will be applied to the red channel.

Green Balance ratio will be applied to the green channel.

Blue Balance ratio will be applied to the blue channel.

6.6.2.5 enum LvBalanceWhiteAuto : LvEnum [strong]

Enum values for the LvDevice::BalanceWhiteAuto feature.

Enumerator

Off Automatic white balance mode off - the automatic white balance is not applied.

Once Automatic white balance mode once - the white balance factors are once adjusted, then switches the enumeration back to the Off value.

Continuous Automatic white balance mode continuous - the white balance is continuously auto-adjusted.

6.6.2.6 enum LvBayerDecoderAlgorithm : LvEnum [strong]

Enum values for the LvDevice::LvBayerDecoderAlgorithm and LvDevice::LvUniBayerDecoderAlgorithm feature.

Enumerator

NearestNeighbour Nearest neighbour algorithm - Fastest decoding, giving the worst results, enables also decoding to a monochrome pixel format.

BilinearInterpolation Bilinear interpolation algorithm - Fast common decoding, enables also decoding to a monochrome pixel format.

BilinearColorCorrection Bilinear color correction algorithm - Decoding with quick enhacements on edges.

PixelGrouping Pixel grouping algorithm - Slower decoding, giving very good results.

VariableGradient Variable gradient algorithm - Slowest decoding, giving the best results.

6.6.2.7 enum LvBlackLevelAuto : LvEnum [strong]

Enum values for the LvDevice::BlackLevelAuto feature.

Enumerator

Off Automatic black level mode off - the black level value is controlled 'manually'.

Once Automatic black level mode 'once' - the black level value is calculated and applied once and the feature switches back to 'off' (manual mode).

Continuous Continuous automatic black level mode - the automatic black level is applied continuously.

6.6.2.8 enum LvBlackLevelSelector : LvEnum [strong]

Enum values for the LvDevice::BlackLevelSelector feature.

Enumerator

All Apply black level on all channels and taps.

6.6.2.9 enum LvBootSwitch : LvEnum [strong]

Enum values for the LvDevice::LvBootSwitch feature.

Enumerator

PureGEV Selects the pure GigE Vision mode strictly following the GigE Vision specification

Legacy Selects the legacy mode allowing dual operation through GigE Vision or custom protocol.

6.6.2.10 enum LvChunkGainSelector : LvEnum [strong]

Enum values for the LvDevice::ChunkGainSelector feature.

Enumerator

AnalogAll Analog gain.

DigitalAll Digital gain.

6.6.2.11 enum LvChunkLvExternalADCSelector : LvEnum [strong]

Enum values for the LvDevice::ChunkLvExternalADCSelector feature.

Enumerator

ExternalADC1 External ADC 1.

ExternalADC2 External ADC 2.

ExternalADC3 External ADC 3.

ExternalADC4 External ADC 4.

6.6.2.12 enum LvChunkSelector : LvEnum [strong]

Enum values for the LvDevice::ChunkSelector feature.

Enumerator

OffsetX Selects the X offset chunk for configuration.
OffsetY Selects the Y offset chunk for configuration.
Width Selects the width chunk for configuration.
Height Selects the height chunk for configuration.
PixelFormat Selects the pixel format chunk for configuration.
LinePitch Selects the line pitch chunk for configuration.
FrameID Selects the frame id chunk for configuration.
Timestamp Selects the time stamp chunk for configuration.
ExposureTime Selects the exposure time chunk for configuration.
Gain Selects the gain chunk for configuration.
LineStatusAll Selects the line status all chunk for configuration.
BlackLevel Selects the black level chunk for configuration.
LvExternalADCValue Selects the external ADC chunk for configuration.
LvSmartAppString Selects the smart application string chunk for configuration.
LvSmartAppInt Selects the smart application signed integer chunk for configuration.
LvSmartAppUint Selects the smart application unsigned integer chunk for configuration.
LvSmartAppRegister Selects the smart application raw register chunk for configuration.
LvTriggerDelayed Selects the trigger delayed chunk for configuration.
LvStrobeDropped Selects the strobe dropped chunk for configuration.
LvFrameAbort Selects the frame abort chunk for configuration.
LvTriggerDropped Selects the trigger dropped chunk for configuration.
LvTriggerError Selects the trigger error chunk for configuration.
LvEncoderPosition Selects the encoder position chunk for configuration.
LvEncoderRotation Selects the encoder rotation chunk for configuration.

6.6.2.13 enum LvColorTransformationSelector : LvEnum [strong]

Enum values for the LvDevice::ColorTransformationSelector feature.

Enumerator

RGBtoRGB RGB to RGB matrix transformation.

6.6.2.14 enum LvColorTransformationValueSelector : LvEnum [strong]

Enum values for the LvDevice::ColorTransformationValueSelector feature.

Enumerator

Gain00 Selects the gain 00 (RR, red-red) entry of the color transformation matrix.
Gain01 Selects the gain 01 (RG, red-green) entry of the colortransformation matrix.
Gain02 Selects the gain 02 (RB, red-blue) entry of the color transformation matrix.
Gain10 Selects the gain 10 (GR, green-red) entry of the color transformation matrix.
Gain11 Selects the gain 11 (GG, green-green) entry of the color transformation matrix.
Gain12 Selects the gain 12 (GB, green-blue) entry of the color transformation matrix.
Gain20 Selects the gain 20 (BR, blue-red) entry of the color transformation matrix.
Gain21 Selects the gain 21 (BG, blue-green) entry of the color transformation matrix.
Gain22 Selects the gain 22 (BB, blue-blue) entry of the color transformation matrix.

6.6.2.15 enum LvCounterEventSource : LvEnum [strong]

Enum values for the LvDevice::CounterEventSource feature.

Enumerator

Off Switches counter event signal off - no signal will be incrementing the counter

FrameTrigger Switches counter event signal to frame trigger - activation of the frame trigger internal signal (before counting down eventual trigger delay) increments the counter.

TimerTick Switches counter event signal to timer tick - 1MHz clock increments the counter.

Line1 Switches counter event signal to line 1 (optocoupler input) - active edge of line 1 increments the counter.

Line2 Switches counter event signal to line 2 (optocoupler input) - active edge of line 2 increments the counter.

Line3 Switches counter event signal to line 3 (optocoupler input) - active edge of line 3 increments the counter.

Line4 Switches counter event signal to line 4 (optocoupler input) - active edge of line 4 increments the counter.

Line17 Switches counter event signal to line 17 (TTL input) - active edge of line 17 increments the counter.

Line18 Switches counter event signal to line 18 (TTL input) - active edge of line 18 increments the counter.

6.6.2.16 enum LvCounterMode : LvEnum [strong]

Enum values for the LvDevice::LvCounterMode feature.

Enumerator

Autoreset Automatic reset mode. Once completed, the counter automatically resets itself and starts counting again.

6.6.2.17 enum LvCounterSelector : LvEnum [strong]

Enum values for the LvDevice::CounterSelector feature.

Enumerator

Counter1 Selects counter 1 for configuration.

Counter2 Selects counter 2 for configuration.

Counter3 Selects counter 3 for configuration.

Counter4 Selects counter 4 for configuration.

6.6.2.18 enum LvDeviceAccess : LvEnum [strong]

This enum is used for opening the Device - see LvDeviceOpen().

Enumerator

None Represents the GenTL DEVICE_ACCESS_NONE. This either means that the Device is not open because it was not opened before or the access to it was denied.

ReadOnly Represents the GenTL DEVICE_ACCESS_READONLY. Open the Device read only. All Port functions can only read from the Device.

Control Represents the GenTL DEVICE_ACCESS_CONTROL. Open the Device in a way that other hosts/processes can have read only access to the Device. Device access level is read/write for this process.

Exclusive Represents the GenTL DEVICE_ACCESS_EXCLUSIVE. Open the Device in a way that only this host/process can have access to the Device. Device access level is read/write for this process.

6.6.2.19 enum LvDeviceAccessStatus : LvEnum [strong]

Values for the LvFtrInfo::DeviceAccessStatus and LvInterface::DeviceAccessStatus features.

Enumerator

Unknown Represents the GenTL DEVICE_ACCESS_STATUS_UNKNOWN. The current availability of the Device is unknown.

ReadWrite Represents the GenTL DEVICE_ACCESS_STATUS_READWRITE - The Device is available for Read/Write.

ReadOnly Represents the GenTL DEVICE_ACCESS_STATUS_READONLY - The Device is available only for Read access (cannot be controlled).

NoAccess Represents the GenTL DEVICE_ACCESS_STATUS_NOACCESS - The Device is not available either because it is already open or because it is not reachable.

6.6.2.20 enum LvDeviceClockSelector : LvEnum [strong]

Enum values for the LvDevice::DeviceClockSelector feature.

Enumerator

SensorDigitization Sensor digitization clock.

6.6.2.21 enum LvDeviceEndianessMechanism : LvEnum [strong]

Enum values for the LvDevice::DeviceEndianessMechanism feature.

Enumerator

Legacy Legacy endianess handling mode, intended for GigE Vision remote devices using GenICam schema version 1.0.

Standard Standard endianess handling mode, intended for GigE Vision remote devices using GenICam schema version 1.1 and newer.

6.6.2.22 enum LvDeviceScanType : LvEnum [strong]

Enum values for the LvDevice::DeviceScanType feature.

Enumerator

Areascan Indicates area scan sensor.

Linescan Indicates line scan sensor.

6.6.2.23 enum LvDeviceTemperatureSelector : LvEnum [strong]

Enum values for the LvDevice::DeviceTemperatureSelector feature.

Enumerator

Sensor Temperature on sensor

Mainboard Temperature on main board

6.6.2.24 enum LvDeviceType : LvEnum [strong]

Enum values for the LvDevice::DeviceType feature.

Enumerator

Custom Device based on a custom technology.

GEV GigE Vision compatible device.

6.6.2.25 enum LvEventNotification : LvEnum [strong]

Enum values for the LvDevice::EventNotification feature.

Enumerator

Off The notifications for the selected event are deactivated.

On The notifications for the selected event are activated.

6.6.2.26 enum LvEventSelector : LvEnum [strong]

Enum values for the LvDevice::EventSelector feature.

Enumerator

LvLog This enumeration value selects the log event for configuration.

LvSmartAppLog This enumeration value selects the smart application log event for configuration.

LvSmartAppString This enumeration value selects the smart application string event for configuration.

LvSmartAppInt This enumeration value selects the smart application signed integer event for configuration.

LvSmartAppUint This enumeration value selects the smart application unsigned integer event for configuration.

LvSmartAppRegister This enumeration value selects the smart application raw register event for configuration.

LvTriggerDropped This enumeration value selects the dropped trigger event for configuration.

6.6.2.27 enum LvExposureAuto : LvEnum [strong]

Enum values for the LvDevice::ExposureAuto feature.

Enumerator

Off Automatic exposure mode off - the automatic exposure is not applied.

Once Automatic exposure mode once - the exposure time is once adjusted, then switches back to off.

Continuous Automatic exposure mode continuous - the exposure time is continuously auto-adjusted.

6.6.2.28 enum LvExposureMode : LvEnum [strong]

Enum values for the LvDevice::ExposureMode feature.

Enumerator

Timed Timed exposure mode - the exposure time is controlled by corresponding feature.

6.6.2.29 enum LvExternalADCSelector : LvEnum [strong]

Enum values for the LvDevice::LvExternalADCSelector feature.

Enumerator

ExternalADC1 Selects external ADC 1 for configuration.

ExternalADC2 Selects external ADC 2 for configuration.

ExternalADC3 Selects external ADC 3 for configuration.

ExternalADC4 Selects external ADC 4 for configuration.

6.6.2.30 enum LvExternalDeviceControlMode : LvEnum [strong]

Enum values for the LvDevice::LvExternalDeviceControlMode feature.

Enumerator

Custom Selects the custom mode.

6.6.2.31 enum LvGainAuto : LvEnum [strong]

Enum values for the LvDevice::GainAuto feature.

Enumerator

Off Automatic gain mode off - the gain value is controlled 'manually'.

Once Automatic gain mode 'once' - the gain value is calculated and applied once and the feature switches back to 'off' (manual mode).

Continuous Continuous automatic gain mode - the AGC is applied continuously.

6.6.2.32 enum LvGainSelector : LvEnum [strong]

Enum values for the LvDevice::GainSelector feature.

Enumerator

All Apply gain on all channels and taps.

AnalogAll Apply analog gain.

DigitalAll Apply digital gain.

6.6.2.33 enum LvGevCCP : LvEnum [strong]

Enum values for the LvDevice::GevCCP feature.

Enumerator

OpenAccess Sets the control channel privilege feature to open.

ExclusiveAccess Sets the control channel privilege feature to exclusive.

ControlAccess Sets the control channel privilege feature to control.

ControlAccessSwitchoverActive Sets the control channel privilege feature to control with switchover active.

6.6.2.34 enum LvGevDeviceClass : LvEnum [strong]

Enum values for the LvDevice::GevDeviceClass feature.

Enumerator

Transmitter Indicates that the device is a GigE Vision transmitter.

6.6.2.35 enum LvGevDeviceModeCharacterSet : LvEnum [strong]

Enum values for the LvDevice::GevDeviceModeCharacterSet feature.

Enumerator

UTF8 Indicates that the camera uses the UTF8 character set.

6.6.2.36 enum LvGevDeviceStreamCaptureMode : LvEnum [strong]

Enum values for the LvDevice::LvGevDeviceStreamCaptureMode feature.

Enumerator

SystemDefault System default mode, configurable in the ini file.

Socket Socket mode, the GVSP stream is processed through the socket interface (regular operating system networks stack).

FilterDriver Filter driver mode, the GVSP stream is processed through the filter driver interface (bypassing operating system network stack).

6.6.2.37 enum LvGevIPConfigurationStatus : LvEnum [strong]

Enum values for the LvDevice::GevIPConfigurationStatus feature.

Enumerator

None No IP configuration method was executed or it is not known.

PersistentIP The current device IP configuration was obtained through persistent IP.

DHCP The current device IP configuration was obtained through DHCP.

LLA The current device IP configuration was obtained through LLA.

ForceIP The current device IP configuration was obtained through ForceIP.

6.6.2.38 enum LvGevSCPDirection : LvEnum [strong]

Enum values for the LvDevice::GevSCPDirection feature.

Enumerator

Transmitter Indicates that the stream channel is a transmitter.

6.6.2.39 enum LvGevSupportedOptionSelector : LvEnum [strong]

Enum values for the LvDevice::GevSupportedOptionSelector feature.

Enumerator

IPConfigurationLLA Indicates whether the (first) network interface supports auto IP addressing (also known as LLA).

IPConfigurationDHCP Indicates whether the (first) network interface supports DHCP IP addressing.

IPConfigurationPersistentIP Indicates whether the (first) network interface supports fixed IP addressing (also known as persistent IP addressing).

CommandsConcatenation Indicates whether command concatenation is supported by the device.

WriteMem Indicates whether write memory scheme is supported by the device.

PacketResend Indicates whether packet resendig is supported by the device.

Event Indicates whether event (message channel) is supported by the device.

EventData Indicates whether eventdata (message channel) is supported by the device.

PendingAck Indicates whether pending acknowledge is supported by the device.

Action Indicates whether action commands are supported by the device.

PrimaryApplicationSwitchover Indicates whether primary application switchover is supported by the device.

ExtendedStatusCodes Indicates whether extended GigE Vision status codes are supported by the device.

DiscoveryAckDelayWritable Indicates whether writable discovery acknowledge delay is supported by the device.

DiscoveryAckDelay Indicates whether discovery acknowledge delay is supported by the device.

TestData Indicates whether test data is supported by the device.

ManifestTable Indicates whether manifest table is supported by the device.

CCPApplicationSocket Indicates whether the primary application port and IP address features are supported by the device.

LinkSpeed Indicates whether link speed feature is supported by the device.

HeartbeatDisable Indicates whether heartbeat disabling is supported by the device.

SerialNumber Indicates whether serial number feature is supported by the device.

UserDefinedName Indicates whether user defined name is supported by the device.

StreamChannelSourceSocket Indicates whether the stream channel source port feature is supported by the device.

StreamChannel0ExtendedChunkData Indicates whether the extended chunk data is supported by the device.

StreamChannel0UnconditionalStreaming Indicates whether the unconditional streaming is supported by the device.

StreamChannel0IPReassembly Indicates whether the reassembly of fragmented IP packets is supported by the device.

StreamChannel0BigAndLittleEndian Indicates whether the big and little endian stream channel is supported by the device.

MessageChannelSourceSocket Indicates whether the message channel source port feature is supported by the device.

6.6.2.40 enum LvImageStampSelector : LvEnum [strong]

Enum values for the LvDevice::LvImageStampSelector feature.

Enumerator

Timestamp Selects the flag controling reset of the image timestamp

FrameID Selects the flag controling reset of the image frame ID

6.6.2.41 enum LvInterfaceType : LvEnum [strong]

Enum values for the LvInterface::InterfaceType feature.

Enumerator

Custom Interface supporting a custom technology devices.

GEV Interface supporting GigE Vision devices.

6.6.2.42 enum LvLensControlCalibrationStatus : LvEnum [strong]

Enum values for the LvDevice::LvLensControlCalibrationStatus feature.

Enumerator

Invalid Current calibration parameters are invalid.

Valid Current calibration parameters are valid.

6.6.2.43 enum LvLensControlTargetApproach : LvEnum [strong]

Enum values for the LvDevice::LvLensControlTargetApproach feature.

Enumerator

Direct Approaches the target position directly, no matter from which side.

FromPlus Approaches the target position always from the plus side to improve accuracy.

FromMinus Approaches the target position always from the minus side to improve accuracy.

6.6.2.44 enum LvLineFormat : LvEnum [strong]

Enum values for the LvDevice::LineFormat feature.

Enumerator

NoConnect Not connected line.

TriState The Line is currently in Tri-state mode (Not driven).

TTL The Line is currently accepting or sending TTL level signals.

LVDS The Line is currently accepting or sending LVDS level signals.

RS422 The Line is currently accepting or sending RS-422 level signals.

OptoCoupled Optically isolated line (optocoupler).

6.6.2.45 enum LvLineMode : LvEnum [strong]

Enum values for the LvDevice::LineMode feature.

Enumerator

Input The line is used as signal input.

Output The line is used as signal output.

6.6.2.46 enum LvLineSelector : LvEnum [strong]

Enum values for the LvDevice::LineSelector feature.

Enumerator

Line1 Selects device's logical line 1 (optocoupler input).

Line2 Selects device's logical line 2 (optocoupler input).

Line3 Selects device's logical line 3 (optocoupler input).

Line4 Selects device's logical line 4 (optocoupler input).

Line5 Selects device's logical line 5.

Line6 Selects device's logical line 6.

Line7 Selects device's logical line 7.

Line8 Selects device's logical line 8.

Line9 Selects device's logical line 9 (optocoupler output).

Line10 Selects device's logical line 10 (optocoupler output).

Line11 Selects device's logical line 11 (optocoupler output).

Line12 Selects device's logical line 12 (optocoupler output).

Line13 Selects device's logical line 13.

Line14 Selects device's logical line 14.

Line15 Selects device's logical line 15.

Line16 Selects device's logical line 16.

Line17 Selects device's logical line 17 (TTL input).

Line18 Selects device's logical line 18 (TTL input).

Line19 Selects device's logical line 19.

Line20 Selects device's logical line 20.

Line21 Selects device's logical line 21.

Line22 Selects device's logical line 22.

Line23 Selects device's logical line 23.

Line24 Selects device's logical line 24.

Line25 Selects device's logical line 25 (TTL output).

Line26 Selects device's logical line 26 (TTL output).

Line27 Selects device's logical line 27.

Line28 Selects device's logical line 286.

Line29 Selects device's logical line 29.

Line30 Selects device's logical line 30.

Line31 Selects device's logical line 31.

Line32 Selects device's logical line 32.

6.6.2.47 enum LvLineSource : LvEnum [strong]

Enum values for the LvDevice::LineSource feature.

Enumerator

Off Switches the line source off. This disables the line output (disconnects the line).

ExposureActive Selects exposure active signal as line source.

Timer1Active Selects timer 1 active signal as line source.

Timer2Active Selects timer 2 active signal as line source.

Timer3Active Selects timer 3 active signal as line source.

Timer4Active Selects timer 4 active signal as line source.

UserOutput1 Selects user output 1 value signal as line source.

UserOutput2 Selects user output 2 value signal as line source.

UserOutput3 Selects user output 3 value signal as line source.

UserOutput4 Selects user output 4 value signal as line source.

UserOutput5 Selects user output 5 value signal as line source.

UserOutput6 Selects user output 6 value signal as line source.

UserOutput7 Selects user output 7 value signal as line source.

UserOutput8 Selects user output 8 value signal as line source.

Counter1 Selects counter 1 active signal as line source.

Counter2 Selects counter 2 active signal as line source.

Counter3 Selects counter 3 active signal as line source.

Counter4 Selects counter 4 active signal as line source.

6.6.2.48 enum LvLUTMode : LvEnum [strong]

Enum values for the LvDevice::LvLUTMode feature.

Enumerator

Direct In this mode the LUT is controlled directly.

BalanceWhite In this mode the LUT is controlled through the higher level features, such as brightness, contrast, gamma or white balance.

6.6.2.49 enum LvLUTSelector : LvEnum [strong]

Enum values for the LvDevice::LUTSelector feature.

Enumerator

Luminance Selects the luminance LUT for configuration.

Red Selects the red LUT for configuration.

Green Selects the green LUT for configuration.

Blue Selects the blue LUT for configuration.

6.6.2.50 enum LvPixelFormat : LvEnum [strong]

LvPixelFormat constants Enum for the LvDevice::PixelFormat, LvDevice::ChunkPixelFormat and LvDevice::LvUniPixelFormat features. The Pixel format constants are defined by the GenICam standard. The value consists of 3 parts:

- byte 1 - color/mono/custom
- byte 2 - bits per pixel
- byte 3 and 4 - ID of the pixel format
  Exceptions are:
- LvPixelFormat::BGR555Packed - used only in the Image Processing Library (in conversions for display).

Enumerator

Mono8 Monochrome 8-bit. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY8BIT | 0x0001).

Mono8S Monochrome 8-bit signed. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY8BIT | 0x0002).

Mono10 Monochrome 10-bit. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0003).

Mono10Packed Monochrome 10-bit packed. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY12BIT | 0x0004).

Mono12 Monochrome 12-bit. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0005).

Mono12Packed Monochrome 12-bit packed. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY12BIT | 0x0006).

Mono14 Monochrome 14-bit. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0025).

Mono16 Monochrome 16-bit. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0007).

BayerGR8 Undecoded 8-bit Bayer array with the GR array position. Defined as (LV_PIX_MONO | LV_PIX↩ _OCCUPY8BIT | 0x0008).

BayerRG8 Undecoded 8-bit Bayer array with the RG array position. Defined as (LV_PIX_MONO | LV_PIX↩ _OCCUPY8BIT | 0x0009).

BayerGB8 Undecoded 8-bit Bayer array with the GB array position. Defined as (LV_PIX_MONO | LV_PIX_↩ OCCUPY8BIT | 0x000A).

BayerBG8 Undecoded 8-bit Bayer array with the BG array position. Defined as (LV_PIX_MONO | LV_PIX_↩ OCCUPY8BIT | 0x000B).

BayerGR10 Undecoded 10-bit Bayer array with the GR array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x000C).

BayerRG10 Undecoded 10-bit Bayer array with the RG array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x000D).

BayerGB10 Undecoded 10-bit Bayer array with the GB array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x000E).

BayerBG10 Undecoded 10-bit Bayer array with the BG array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x000F).

BayerGR12 Undecoded 12-bit Bayer array with the GR array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x0010).

BayerRG12 Undecoded 12-bit Bayer array with the RG array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x0011).

BayerGB12 Undecoded 12-bit Bayer array with the GB array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x0012).

BayerBG12 Undecoded 12-bit Bayer array with the BG array position. Defined as (LV_PIX_MONO | LV_PI↩ X_OCCUPY16BIT | 0x0013).

BayerGR10Packed Undecoded 10-bit packed Bayer array with the GR array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x0026).

BayerRG10Packed Undecoded 10-bit packed Bayer array with the RG array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x0027).

BayerGB10Packed Undecoded 10-bit packed Bayer array with the GB array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x0028).

BayerBG10Packed Undecoded 10-bit packed Bayer array with the BG array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x0029).

BayerGR12Packed Undecoded 12-bit packed Bayer array with the GR array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x002A).

BayerRG12Packed Undecoded 12-bit packed Bayer array with the RG array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x002B).

BayerGB12Packed Undecoded 12-bit packed Bayer array with the GB array position. Defined as (LV_PIX↩ _MONO | LV_PIX_OCCUPY12BIT | 0x002C).

BayerBG12Packed Undecoded 10-bit packed Bayer array with the BG array position. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY12BIT | 0x002D).

BayerGR16 Undecoded 16-bit Bayer array with the GR array position. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x002E).

BayerRG16 Undecoded 16-bit Bayer array with the RG array position. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x002F).

BayerGB16 Undecoded 16-bit Bayer array with the GB array position. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0030).

BayerBG16 Undecoded 16-bit Bayer array with the BG array position. Defined as (LV_PIX_MONO | LV_PIX_OCCUPY16BIT | 0x0031).

RGB8 RGB 24-bit packed (3x8 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY24BIT | 0x0014).

BGR8 BGR 24-bit packed (3x8 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY24BIT | 0x0015).

RGBA8 RGB 32-bit packed (3x8 bits + 1x8 bits alpha). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY32BIT | 0x0016).

BGRA8 BGR 32-bit packed (3x8 bits + 1x8 bits alpha). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY32BIT | 0x0017).

RGB10 RGB 48-bit packed (3x10 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0018).

BGR10 BGR 48-bit packed (3x10 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0019).

RGB12 RGB 48-bit packed (3x12 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x001A).

BGR12 BGR 48-bit packed (3x12 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x001B).

RGB16 RGB 48-bit packed (3x16 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0033).

RGB10V1Packed RGB 32-bit packed (3x10 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY32BIT | 0x001C).

RGB10P32 RGB 32-bit packed (3x10 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY32BIT | 0x001D).

RGB12V1Packed RGB 36-bit packed (3x12 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY36BIT | 0x0034).

RGB565P RGB 16-bit packed (5,6,5 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x0035).

BGR565P BGR 16-bit packed (5,6,5 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x0036).

YUV411_8 YUV 4-1-1 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY12BIT | 0x001E).

YUV422_8_UYVY YUV 4-2-2 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x001F).

YUV422_8 YUV 4-2-2 YUYV Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x0032).

YUV8 YUV 4-4-4 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY24BIT | 0x0020).

YCbCr422_8 YCbCr 4-2-2 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x003B).

YCbCr601_422_8 YCbCr 4-2-2 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x003B).

YCbCr601_422_8_CbYCrY YCbCr 4-2-2 Packed. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x003B).

RGB8_Planar RGB 8-bit planar. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY24BIT | 0x0021).

RGB10_Planar RGB 10-bit planar. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0022).

RGB12_Planar RGB 12-bit planar. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0023).

RGB16_Planar RGB 16-bit planar. Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY48BIT | 0x0024).

BGR555P RGB 15-bit packed (3x5 bits). Defined as (LV_PIX_COLOR | LV_PIX_OCCUPY16BIT | 0x00E1). Not a standard GenICam format, used only in the image processing library.

Mono8Signed Alias for LvPixelFormat::Mono8S.

RGB8Packed Alias for LvPixelFormat::RGB8.

BGR8Packed Alias for LvPixelFormat::BGR8.

RGBA8Packed Alias for LvPixelFormat::RGBA8.
BGRA8Packed Alias for LvPixelFormat::BGRA8.
RGB10Packed Alias for LvPixelFormat::RGB10.
BGR10Packed Alias for LvPixelFormat::BGR10.
RGB12Packed Alias for LvPixelFormat::RGB12.
BGR12Packed Alias for LvPixelFormat::BGR12.
RGB16Packed Alias for LvPixelFormat::RGB16.
RGB10V2Packed Alias for LvPixelFormat::RGB10P32.
RGB565Packed Alias for LvPixelFormat::RGB565P.
BGR565Packed Alias for LvPixelFormat::BGR565P.
YUV411Packed Alias for LvPixelFormat::YUV411_8.
YUV422Packed Alias for LvPixelFormat::YUV422_8_UYVY.
YUV422YUYVPacked Alias for LvPixelFormat::YUV422_8.
YUV444Packed Alias for LvPixelFormat::YUV8.
RGB8Planar Alias for LvPixelFormat::RGB8_Planar.
RGB10Planar Alias for LvPixelFormat::RGB10_Planar.
RGB12Planar Alias for LvPixelFormat::RGB12_Planar.
RGB16Planar Alias for LvPixelFormat::RGB16_Planar.
Mono8s Alias for LvPixelFormat::Mono8S.
RGBa8 Alias for LvPixelFormat::RGBA8.
BGRa8 Alias for LvPixelFormat::BGRA8.
RGB565p Alias for LvPixelFormat::RGB565P.
BGR565p Alias for LvPixelFormat::BGR565P.
RGB10p32 Alias for LvPixelFormat::RGB10P32.
BGR555p Alias for LvPixelFormat::BGR555P.
YUV411_8_UYYVYY Alias for LvPixelFormat::YUV411_8.
YUV8_UYV Alias for LvPixelFormat::YUV8.

6.6.2.51 enum LvPowerSwitchBoundADC : LvEnum [strong]

Enum values for the LvDevice::LvPowerSwitchBoundADC feature.

Enumerator

None Binds no external ADC to the power switch
ExternalADC1 Binds external ADC 1 to the power switch
ExternalADC2 Binds external ADC 2 to the power switch
ExternalADC3 Binds external ADC 3 to the power switch
ExternalADC4 Binds external ADC 4 to the power switch

6.6.2.52 enum LvPowerSwitchCurrentAction : LvEnum [strong]

Enum values for the LvDevice::LvPowerSwitchCurrentAction feature.

Enumerator

Idle Reports that all power switches are idle
Pulse Reports that a pulse command is pending
Calibrate Reports that a calibration is pending
AdjustPosition Reports that a position adjustment is pending
Drive Reports that a power switch drive operation is pending

6.6.2.53 enum LvPowerSwitchDrive : LvEnum [strong]

Enum values for the LvDevice::LvPowerSwitchDrive feature.

Enumerator

Off Switches the selected power switch off.

Plus Sswitches the selected power switch to plus polarity.

Minus Switches the selected power switch to minus polarity.

6.6.2.54 enum LvPowerSwitchDriveAll : LvEnum [strong]

Enum values for the LvPowerSwitchDriveAll feature.

Enumerator

Off Switches the active power switches off

Plus Switches the active power switches to plus polarity

Minus Switches the active power switches to minus polarity

6.6.2.55 enum LvPowerSwitchSelector : LvEnum [strong]

Enum values for the LvDevice::LvPowerSwitchSelector feature.

Enumerator

PowerSwitch1 Selects power switch 1 for configuration.

PowerSwitch2 Selects power switch 2 for configuration.

PowerSwitch3 Selects power switch 3 for configuration.

PowerSwitch4 Selects power switch 4 for configuration.

6.6.2.56 enum LvRegionSelector : LvEnum [strong]

Enum values for the LvDevice::RegionSelector feature.

Enumerator

Region0 Selects region 0 for configuration.

Region1 Selects region 1 for configuration.

Region2 Selects region 2 for configuration.

Region3 Selects region 3 for configuration.

6.6.2.57 enum LvRenderType : LvEnum [strong]

Enum values for the LvRenderer::LvRenderType feature.

Enumerator

FullSize Renders the acquired image in full size.

ScaleToFit Renders the acquired image to fit into the window.

ScaleToSize Renders the acquired image scaled to required size.

ScaleToTiles Renders the acquired images in tiles.

6.6.2.58 enum LvSerialPortBaudRate : LvEnum [strong]

Enum values for the LvDevice::LvSerialPortBaudRate feature.

Enumerator

Baud2400 Baud rate of 2400 bauds.

Baud4800 Baud rate of 4800 bauds.

Baud9600 Baud rate of 9600 bauds.

Baud14400 Baud rate of 14400 bauds.

Baud19200 Baud rate of 19200 bauds.

Baud38400 Baud rate of 38400 bauds.

Baud57600 Baud rate of 57600 bauds.

Baud115200 Baud rate of 115200 bauds.

6.6.2.59 enum LvSerialPortCommandStatus : LvEnum [strong]

Enum values for the LvDevice::LvSerialPortCommandStatus feature.

Enumerator

Success Last command was successfully transferred.

Timeout Last command ended with timeout (depending on configuration this might be problem or not).

PortBusy Last command failed: port busy.

CommunicationError Last command failed: generic communication error.

FrameError Last command failed: frame error.

ParityError Last command failed: parity error.

Overflow Last command failed: overflow.

6.6.2.60 enum LvSerialPortDataBits : LvEnum [strong]

Enum values for the LvDevice::LvSerialPortDataBits feature.

Enumerator

DataBits7 7 data bits per character.

DataBits8 8 data bits per character.

6.6.2.61 enum LvSerialPortParity : LvEnum [strong]

Enum values for the LvDevice::LvSerialPortParity feature.

Enumerator

None Parity method 'none', parity bit not used.

Odd Parity method 'odd', odd number of set bits in each character.

Even Parity method 'even', even number of set bits in each character.

6.6.2.62 enum LvSerialPortStopBits : LvEnum [strong]

Enum values for the LvDevice::LvSerialPortStopBits feature.

Enumerator

StopBits1 1 stop bit per character.

StopBits1dot5 1.5 stop bit per character.

StopBits2 2 stop bits per character.

6.6.2.63 enum LvSpecialPurposeTriggerActivation : LvEnum [strong]

Enum values for the LvDevice::LvSpecialPurposeTriggerActivation feature.

Enumerator

RisingEdge Selects the trigger signal's rising edge as active.

FallingEdge Selects the trigger signal's falling edge as active

6.6.2.64 enum LvSpecialPurposeTriggerSelector : LvEnum [strong]

Enum values for the LvDevice::LvSpecialPurposeTriggerSelector feature.

Enumerator

ImageStampsReset Timestamps reset trigger - controls reset of timestamp, frame ID and other image stamps.

6.6.2.65 enum LvSpecialPurposeTriggerSource : LvEnum [strong]

Enum values for the LvDevice::LvSpecialPurposeTriggerSource feature.

Enumerator

Off Sets trigger source off - it can be still be issued by an explicit software trigger

Line1 Sets the signal source for the selected trigger to line 1 (optocoupler input).

Line2 Sets the signal source for the selected trigger to line 2 (optocoupler input).

Line3 Sets the signal source for the selected trigger to line 3 (optocoupler input).

Line4 Sets the signal source for the selected trigger to line 4 (optocoupler input).

Line5 Sets the signal source for the selected trigger to line 5.

Line6 Sets the signal source for the selected trigger to line 6.

Line7 Sets the signal source for the selected trigger to line 7.

Line8 Sets the signal source for the selected trigger to line 8.

Line17 Sets the signal source for the selected trigger to line 17 (TTL input).

Line18 Sets the signal source for the selected trigger to line 18 (TTL input).

Line19 Sets the signal source for the selected trigger to line 19.

Line20 Sets the signal source for the selected trigger to line 20.

Line21 Sets the signal source for the selected trigger to line 21.

Line22 Sets the signal source for the selected trigger to line 22.

Line23 Sets the signal source for the selected trigger to line 23.

Line24 Sets the signal source for the selected trigger to line 24.

Action1 Sets the signal source for the selected trigger to action signal 1.

Action2 Sets the signal source for the selected trigger to action signal 2.

Action3 Sets the signal source for the selected trigger to action signal 3.

Action4 Sets the signal source for the selected trigger to action signal 4.

Action5 Sets the signal source for the selected trigger to action signal 5.

Action6 Sets the signal source for the selected trigger to action signal 6.

Action7 Sets the signal source for the selected trigger to action signal 7.

Action8 Sets the signal source for the selected trigger to action signal 8.

6.6.2.66 enum LvStreamAcquisitionModeSelector : LvEnum [strong]

Enum values for the LvStream::StreamAcquisitionModeSelector feature.

Enumerator

Default Default acquisition mode.

6.6.2.67 enum LvStreamType : LvEnum [strong]

Enum values for the LvStream::StreamType feature.

Enumerator

Custom Stream belonging to a custom technology device.

GEV Stream belonging to a GigE Vision compatible device.

6.6.2.68 enum LvStrobeDropMode : LvEnum [strong]

Enum values for the LvDevice::LvStrobeDropMode feature.

Enumerator

DropStrobe Strobe drop mode 'drop' - the strobe is dropped, image is acquired without the strobe.

DelayFrame Strobe drop mode 'delay' - the frame acquisition is delayed, until the strobe can be issued.

6.6.2.69 enum LvStrobeDurationMode : LvEnum [strong]

Enum values for the LvDevice::LvStrobeDurationMode feature.

Enumerator

FrameRateRelated The maximum strobe duration depends on the maximum frame rate of the camera. For very fast sensors the max. strobe duration time, dependent on the specification of the LEDs used, cannot be applied in full length, as the recovery time may become too short. The calculation is done automatically depending on the LEDs used and the max. frame rate of the camera in its actual mode of operation. Such calculation also includes boosted frame rates e.g. when the camera is in partial scanning and/or binning mode.

Free The maximum strobe duration depends on the maximum allowed ON-time and the minimum required recovery time of the LEDs used. The user can program the strobe duration free, according to his request, but must be aware himself about the relation of strobe ON-time and recovery time. An automatic protection circuit in HW drops a strobe in case the proper relation of ON-time to recovery time is not guaranteed. In such case a related error code is returned by the SW (frame message or otherwise)

6.6.2.70 enum LvStrobeEnable : LvEnum [strong]

Enum values for the LvDevice::LvStrobeEnable feature.

Enumerator

Off Switches the strobe off.

AllClusters Switches on all LED clusters of the strobe light. On strobe lights possessing just a single LED cluster this cluster is switche on.

LEDCluster1 Switches on LED cluster 1 only. The strobe will use just the LEDs in this cluster.

LEDCluster2 Switches on LED cluster 2 only. The strobe will use just the LEDs in this cluster.

6.6.2.71 enum LvTimerSelector : LvEnum [strong]

Enum values for the LvDevice::TimerSelector feature.

Enumerator

Timer1 Selects timer 1 for configuration.

Timer2 Selects timer 2 for configuration.

Timer3 Selects timer 3 for configuration.

Timer4 Selects timer 4 for configuration.

6.6.2.72 enum LvTimerTriggerSource : LvEnum [strong]

Enum values for the LvDevice::TimerTriggerSource feature.

Enumerator

Off Switches timer trigger signal off - no signal will be firing the timer

FrameTrigger Switches timer trigger signal to frame trigger - activation of the frame trigger internal signal (before counting down eventual trigger delay) activates the timer.

Counter1End Switches timer trigger signal to counter 1 end - expiration of counter 1 activates the timer.

Counter2End Switches timer trigger signal to counter 2 end - expiration of counter 2 activates the timer.

Counter3End Switches timer trigger signal to counter 3 end - expiration of counter 3 activates the timer.

Counter4End Switches timer trigger signal to counter 4 end - expiration of counter 4 activates the timer.

UserOutput1 Switches timer trigger signal to user output 1 - activation of user output 1 activates the timer.

UserOutput2 Switches timer trigger signal to user output 2 - activation of user output 2 activates the timer.

UserOutput3 Switches timer trigger signal to user output 3 - activation of user output 3 activates the timer.

UserOutput4 Switches timer trigger signal to user output 4 - activation of user output 4 activates the timer.

UserOutput5 Switches timer trigger signal to user output 5 - activation of user output 5 activates the timer.

UserOutput6 Switches timer trigger signal to user output 6 - activation of user output 6 activates the timer.

UserOutput7 Switches timer trigger signal to user output 7 - activation of user output 7 activates the timer.

UserOutput8 Switches timer trigger signal to user output 8 - activation of user output 8 activates the timer.

6.6.2.73 enum LvTLType : LvEnum [strong]

Enum values for the LvSystem::TLType feature.

Enumerator

Mixed GenTL producer supporting mixed technologies.

Custom GenTL producer supporting a custom technology devices.

GEV GenTL producer supporting GigE Vision devices.

6.6.2.74 enum LvTriggerActivation : LvEnum [strong]

Enum values for the LvDevice::TriggerActivation feature.

Enumerator

RisingEdge Selects the trigger signal's rising edge as active.

FallingEdge Selects the trigger signal's falling edge as active

6.6.2.75 enum LvTriggerCaching : LvEnum [strong]

Enum values for the LvDevice::LvTriggerCaching feature.

Enumerator

Cache Trigger caching mode 'cache' - early triggers are cached and applied as soon as possible.

Drop Trigger caching mode 'cache' - early triggers are dropped

6.6.2.76 enum LvTriggerMode : LvEnum [strong]

Enum values for the LvDevice::TriggerMode feature.

Enumerator

Off Trigger mode off - disables selected trigger

On Trigger mode on - enables selected trigger.

6.6.2.77 enum LvTriggerSelector : LvEnum [strong]

Enum values for the LvDevice::TriggerSelector feature.

Enumerator

FrameStart Frame start trigger - controls a new frame acquisition.

FrameBurstStart Frame burst start trigger - Selects a trigger starting the capture of the bursts of frames.

6.6.2.78 enum LvTriggerSource : LvEnum [strong]

Enum values for the LvDevice::TriggerSource feature.

Enumerator

Line1 Sets the signal source for the selected trigger to line 1 (optocoupler input).

Line2 Sets the signal source for the selected trigger to line 2 (optocoupler input).

Line3 Sets the signal source for the selected trigger to line 3 (optocoupler input).

Line4 Sets the signal source for the selected trigger to line 4 (optocoupler input).

Line5 Sets the signal source for the selected trigger to line 5.

Line6 Sets the signal source for the selected trigger to line 6.

Line7 Sets the signal source for the selected trigger to line 7.

Line8 Sets the signal source for the selected trigger to line 8.

Line17 Sets the signal source for the selected trigger to line 17 (TTL input).

Line18 Sets the signal source for the selected trigger to line 18 (TTL input).

Line19 Sets the signal source for the selected trigger to line 19.

Line20 Sets the signal source for the selected trigger to line 20.

Line21 Sets the signal source for the selected trigger to line 21.

Line22 Sets the signal source for the selected trigger to line 22.

Line23 Sets the signal source for the selected trigger to line 23.

Line24 Sets the signal source for the selected trigger to line 24.

Software Sets the signal source for the selected trigger to software.

Action1 Sets the signal source for the selected trigger to action signal 1.

Action2 Sets the signal source for the selected trigger to action signal 2.

Action3 Sets the signal source for the selected trigger to action signal 3.

Action4 Sets the signal source for the selected trigger to action signal 4.

Action5 Sets the signal source for the selected trigger to action signal 5.

Action6 Sets the signal source for the selected trigger to action signal 6.

Action7 Sets the signal source for the selected trigger to action signal 7.

Action8 Sets the signal source for the selected trigger to action signal 8.

Quad Sets the signal source for the selected trigger to quadrature decoder.

Counter1 Sets the signal source for the selected trigger to counter 1.

Counter2 Sets the signal source for the selected trigger to counter 2.

Counter3 Sets the signal source for the selected trigger to counter 3.

Counter4 Sets the signal source for the selected trigger to counter 4.

Timer1 Sets the signal source for the selected trigger to timer 1.

Timer2 Sets the signal source for the selected trigger to timer 2.

Timer3 Sets the signal source for the selected trigger to timer 3.

Timer4 Sets the signal source for the selected trigger to timer 4.

6.6.2.79 enum LvUniBalanceRatioSelector : LvEnum [strong]

Enum values for the LvDevice::LvUniBalanceRatioSelector feature.

Enumerator

Red Selects the red channel for configuration.

Green Selects the green channel for configuration.

Blue Selects the blue channel for configuration.

6.6.2.80 enum LvUniBalanceWhiteAuto : LvEnum [strong]

Enum values for the LvDevice::LvUniBalanceWhiteAuto feature.

Enumerator

Off Automatic white balance mode off - the automatic white balance is not applied.

Once Automatic white balance mode once - the white balance factors are once adjusted, then switches back to off.

6.6.2.81 enum LvUniColorTransformationMode : LvEnum [strong]

Enum values for the LvDevice::LvUniColorTransformationMode feature.

Enumerator

Direct In this mode the Color Transformation matrix can be controlled directly.

Generated In this mode the Color Transformation matrix is set through the higher level features, such as the Saturation.

6.6.2.82 enum LvUniColorTransformationSelector : LvEnum [strong]

Enum values for the LvDevice::LvUniColorTransformationSelector feature.

Enumerator

RGBtoRGB RGB to RGB matrix transformation.

6.6.2.83 enum LvUniColorTransformationValueSelector : LvEnum [strong]

Enum values for the LvDevice::LvUniColorTransformationValueSelector feature.

Enumerator

Gain00 Selects the gain 00 (RR, red-red) entry of the color transformation matrix.

Gain01 Selects the gain 01 (RG, red-green) entry of the colortransformation matrix.

Gain02 Selects the gain 02 (RB, red-blue) entry of the color transformation matrix.

Gain10 Selects the gain 10 (GR, green-red) entry of the color transformation matrix.

Gain11 Selects the gain 11 (GG, green-green) entry of the color transformation matrix.

Gain12 Selects the gain 12 (GB, green-blue) entry of the color transformation matrix.

Gain20 Selects the gain 20 (BR, blue-red) entry of the color transformation matrix.

Gain21 Selects the gain 21 (BG, blue-green) entry of the color transformation matrix.

Gain22 Selects the gain 22 (BB, blue-blue) entry of the color transformation matrix.

6.6.2.84 enum LvUniLUTMode : LvEnum [strong]

Enum values for the LvDevice::LvUniLUTMode feature.

Enumerator

Direct In this mode the LUT is controlled directly.

Generated In this mode the LUT is controlled through the higher level features, such as brightness, contrast, gamma or white balance.

6.6.2.85 enum LvUniLUTSelector : LvEnum [strong]

Enum values for the LvDevice::LvUniLUTSelector feature.

Enumerator

Luminance Selects the luminance LUT for configuration.

Red Selects the red LUT for configuration.

Green Selects the green LUT for configuration.

Blue Selects the blue LUT for configuration.

6.6.2.86 enum LvUniProcessExecution : LvEnum [strong]

Enum values for the LvDevice::LvUniProcessExecution feature.

Enumerator

OnBufferPtrQuery The SW image processing is delayed to the time the application asks for the LvBuffer::UniBase or LvBuffer::ProcessBase pointer or for the LvipImgInfo data. This enables to the application to skip the processing in case it is not needed. If this is queried several times for the same image, the processing is done only once.

OnPopFromQueue The SW image processing is done at the moment the buffer is popped from the output buffer queue, before delivering it to the application.

OnExplicitRequest The SW processing is not done automatically, but must be explicitly called by the LvBuffer::ExecProcess.

6.6.2.87 enum LvUniProcessMode : LvEnum [strong]

Enum values for the LvDevice::LvUniProcessMode feature.

Enumerator

HwOnly HwOnly - The processing is done only in case it is available directly on the hardware (device). The images will be delivered to the output buffer queue already processed.

SwOnly SwOnly - The processing will be done by software even if the hardware could support the operation. The software processing is done when the buffer is passed to the output buffer queue (or later - see LvUniProcessExecution).

Auto Auto - The processing will be done by hardware and by software will be processed only the part, which is not possible to do on hardware. Note that if the Bayer decoding is done by software (this happens when you select an undecoded Bayer pixel format as the device PixelFormat), the LUT must be then also done by software, even if it is available in hardware; that's because it must be applied after the Bayer decoding.

Off The automatic processing is not available. You can use the HW features (LUT etc.) directly.

6.6.2.88 enum LvUserOutputSelector : LvEnum [strong]

Enum values for the LvDevice::UserOutputSelector feature.

Enumerator

UserOutput1 Selects user output 1.

UserOutput2 Selects user output 2.

UserOutput3 Selects user output 3.

UserOutput4 Selects user output 4.

UserOutput5 Selects user output 5.

UserOutput6 Selects user output 6.

UserOutput7 Selects user output 7.

UserOutput8 Selects user output 8.

6.6.2.89 enum LvUserSetDefaultSelector : LvEnum [strong]

Enum values for the LvDevice::UserSetDefaultSelector feature.

Enumerator

Default Selects the default user set as the default startup set.

UserSet1 Selects user set 1 as the default startup set.

UserSet2 Selects user set 2 as the default startup set.

UserSet3 Selects user set 3 as the default startup set.

UserSet4 Selects user set 4 as the default startup set.

None When resetting/connecting the camera, no user set is applied, the last camera configuration remains. During camera boot, the default user set is applied.

6.6.2.90 enum LvUserSetSelector : LvEnum [strong]

Enum values for the LvDevice::UserSetSelector feature.

Enumerator

Default Selects the default configuration set.

UserSet1 Selects user set 1.

UserSet2 Selects user set 2.

UserSet3 Selects user set 3.

UserSet4 Selects user set 4.

6.7 SynView Image Processing .Net Class Library Defines

Modules

- Definitions for Enumeration Entry Info
- LvStatus definitions

Classes

- struct LvipImgInfo

Macros

- #define LVIP_LUT_BAYER
- #define LVIP_LUT_BAYER_16

6.7.1 Detailed Description

6.7.2 Macro Definition Documentation

6.7.2.1 #define LVIP_LUT_BAYER

If the LUT is to be used in Bayer decoding/encoding functions, this attribute is to be OR-ed to the LvipLutType specification in the LvipAllocateLut() function Bayer LUT requires bigger size - is needed for the bilinear interpolation methods and for 10- and 12-bit source formats.

6.7.2.2 #define LVIP_LUT_BAYER_16

Bayer16 is a subset of LVIP_LUT_BAYER, suitable for all 10- and 12-bit decoding, with the exception of LvipBd↩ BilinearInterpolation() function.

6.8 SynView Image Processing .Net Class Library Enums

Enumerations

- enum LvipImgAttr : UInt32 {
  None, BottomUp, DWordAligned, QWordAligned,
  SSEAligned, NotDataOwner }
- enum LvipOption : UInt32 {
  None, ReallocateDst, TiffConvertTo16Bit, BmpForceTopDown,
  BmpForceBottomUp, JpegConvertToBgr, JpegReadHeaderOnly, WbCorrectFactors }
- enum LvipLutType : LvEnum {
  Uni, Type8Bit, Type10Bit, Type12Bit,
  UniBayer, Type8BitBayer, Type10BitBayer, Type12BitBayer,
  UniBayer16, Type10BitBayer16, Type12BitBayer16 }
- enum LvipColor : LvEnum { None }
- enum LvipTextAttr : UInt32 {
  None, Bold, Italic, Underline,
  Strikeout, Nonantialiased, Shadow, Outline,
  ShadowRB, ShadowRT, ShadowLB, ShadowLT,
  ShadowB, ShadowT, ShadowR, ShadowL }

6.8.1 Detailed Description

6.8.2 Enumeration Type Documentation

6.8.2.1 enum LvipColor : LvEnum [strong]

Color definitions for the Overlay functions.

Enumerator

None Defines a non-color. This is useful for the transparent color - specifying the transparent color as LvipColor::None in LvipSetOverlayTransparentColor() switches off overlay transparency.

6.8.2.2 enum LvipImgAttr : UInt32 [strong]

Image attributes. Flags to be used in the Attributes of the LvipImgInfo structure.

Enumerator

None To be used when no flags are to be set.

BottomUp Lines in the image buffer are ordered from the bottom line to the top line, so the image bufer begins with the bottom line.

DWordAligned The line increment is aligned to double word (32 bits). This is required for example by the Windows Device Independent Bitmap format (DIB, BMP) This attribute is used only in the LvipInitImgInfo() function (which can be called as a result of the LvipOption::ReallocateDst attribute).

QWordAligned The line increment is aligned to quad word (64 bits). This attribute is used in the LvipInitImgInfo() function (which can be called as a result of the LvipOption::ReallocateDst attribute).

SSEAligned The line increment is aligned to SSE word (128 bits). This attribute is used in the LvipInitImgInfo() function (which can be called as a result of the LvipOption::ReallocateDst attribute).

NotDataOwner The LvipImgInfo is not the owner of image data, so the LvipDeallocateImageData() function will not attempt to deallocate the image data. This attribute is used when the image data are owned by another LvipImgInfo or belonging to other code, for example when working directly with the image in the DMA buffer. Note that LvipDeallocateImageData() may be called from other functions, for example, when you use the LvipOption::ReallocateDst attribute.

6.8.2.3 enum LvipLutType : LvEnum [strong]

LUT type - to be used in the LvipAllocateLut() function.

Enumerator

Uni LUT which internally contains 3 LUTs: 8-bit, 10-bit and 12-bit. All the LUTs are kept synchronized.

Type8Bit 8-bit LUT type, used for images with LvPixelFormat::Mono8, LvPixelFormat::BGR8Packed or LvPixelFormat::BGRA8Packed.

Type10Bit 10-bit LUT type, used for images with LvPixelFormat::Mono10.

Type12Bit 12-bit LUT type, used for images with LvPixelFormat::Mono12

UniBayer LvipLutType::Uni type with the LVIP_LUT_BAYER.

Type8BitBayer LvipLutType::Type8Bit type with the LVIP_LUT_BAYER.

Type10BitBayer LvipLutType::Type10Bit type with the LVIP_LUT_BAYER.

Type12BitBayer LvipLutType::Type12Bit type with the LVIP_LUT_BAYER.

UniBayer16 LvipLutType::Uni type with the LVIP_LUT_BAYER_16.

Type10BitBayer16 LvipLutType::Type10Bit type with the LVIP_LUT_BAYER_16.

Type12BitBayer16 LvipLutType::Type12Bit type with the LVIP_LUT_BAYER_16.

6.8.2.4 enum LvipOption : UInt32 [strong]

Options for image processing functions in the Options parameter.

Enumerator

None To be used when no flags are to be set.

ReallocateDst The destination image data can be reallocated if it is needed. If the function stores a result of the operation to the destination image buffer, it first checks if the destination LvipImgInfo has appropriate parameters and the buffer(s) allocated. If not and this attribute is specified, it adapts the parameters of the LvipImgInfo and reallocates the buffer as needed. If this attribute is not specified, the function returns an error in case of mismatch.

TiffConvertTo16Bit The attribute will force conversion of the image to 16-bit mono format, if it is in 9- to 15-bit mono format. This can be used used when saving mono image to TIFF by the LvipSaveToTiff() function, as many software packages do not understand mono TIFF if it is in 9- to 15-bit mono format.

BmpForceTopDown The BMP file will be read to the top-down line layout. This attribute is used in the LvipLoadFromBmp() and LvipSaveToBmp() functions, as the BMP format can be either in the bottom-up line layout or in the top-down line layout.

BmpForceBottomUp The BMP file will be read to the bottom-up line layout. This attribute is used in the LvipLoadFromBmp() and LvipSaveToBmp() functions, as the BMP format can be either in the bottom-up line layout or in the top-down line layout.

JpegConvertToBgr The color JPEG images are stored in RGB format (24-bit). With this option the pixel format will be reversed to the BGR format in the LvipLoadFromJpg() function.

JpegReadHeaderOnly The JPEG image data will not be read, only the header will be read. This enables to allocate the image buffer and then read the full image.

WbCorrectFactors This attribute can be used in the LvipCalcWbFactors() function. If present, it is assumed that the white balance is calculated from the image to which were applied white balancing factors passed as input parameters. Thus only a correction is calculated and the existing factors are modified.

6.8.2.5 enum LvipTextAttr : UInt32 [strong]

Text attributes definitions for the Overlay functions.

Enumerator

None To be used when no flags are to be set.

Bold Bold text. Text attribute for the LvipSetOverlayTextParams() function: Bold text

Italic Italics text. Text attribute for the LvipSetOverlayTextParams() function: Italics text

Underline Underlined text. Text attribute for the LvipSetOverlayTextParams() function: Underlined text

Strikeout Strikeout text. Text attribute for the LvipSetOverlayTextParams() function: Strikeout text

Nonantialiased Text antialiasing off. Text attribute for the LvipSetOverlayTextParams() function: Text antialiasing will be switched off - this is useful for text on transparent background, where antialiasing (like ClearType) can make undesirable effects.

Shadow Text with a 1 pixel shadow. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at right-bottom direction.

Outline Text with a 1 pixel outline. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel outline around the letters. This is useful namely for the text on transparent background - by adding the outline of different color, then the text is readable even if the background become of the same color, as the text.

ShadowRB Text with a 1 pixel shadow at right-bottom direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at right-bottom direction (equal to LvipTextAttr::Shadow constant). This constant can be combined with other LvipTextAttr::ShadowXX constants.

ShadowRT Text with a 1 pixel shadow at right-top direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at right-top direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowLB Text with a 1 pixel shadow at left-bottom direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at left-bottom direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowLT Text with a 1 pixel shadow at left-top direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at left-top direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowB Text with a 1 pixel shadow at bottom direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at bottom direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowT Text with a 1 pixel shadow at top direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at top direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowR Text with a 1 pixel shadow at right direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at right direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

ShadowL Text with a 1 pixel shadow at left direction. Text attribute for the LvipSetOverlayTextParams() function: Text with a 1 pixel shadow at left direction. This constant can be combined with other LVIP_TEXTATTR_SHADOW_x constants.

6.9 Definitions for Enumeration Entry Info

Macros

- #define LV_ENUMENTRY_CURRENT

6.9.1 Detailed Description

6.9.2 Macro Definition Documentation

6.9.2.1 #define LV_ENUMENTRY_CURRENT

If used as Param of the LvGetInfo(), LvGetInfoStr() and LvGetInfoStrSize() the returned value is for the current enum entry.

6.10 LvStatus definitions

Macros

- #define LVSTATUS_TIMEOUT
- #define LVSTATUS_LVIP_DST_RECT_OUTSIDE_SRC

6.10.1 Detailed Description

Here are only those definitions, the values of which you might need to know in order to test a function result. All the other defines of LvStatus values are available in the sv.synview.status.h file.

6.10.2 Macro Definition Documentation

6.10.2.1 #define LVSTATUS_LVIP_DST_RECT_OUTSIDE_SRC

The specified rectangle is outside of source image data. It means that you are trying to copy an area which not exist in the source image. To create a destination image, the rectangle must at least partially overlap the source image.

6.10.2.2 #define LVSTATUS_TIMEOUT

The function has returned because a timeout has expired.

6.11 LvPixelFormat definitions

Macros

- #define LV_PIX_MONO 0x01000000
- #define LV_PIX_COLOR 0x02000000
- #define LV_PIX_CUSTOM 0x80000000
- #define LV_PIX_COLOR_MASK
- #define LV_PIX_OCCUPY8BIT 0x00080000
- #define LV_PIX_OCCUPY12BIT 0x000C0000
- #define LV_PIX_OCCUPY16BIT 0x00100000
- #define LV_PIX_OCCUPY24BIT 0x00180000
- #define LV_PIX_OCCUPY32BIT 0x00200000
- #define LV_PIX_OCCUPY36BIT 0x00240000
- #define LV_PIX_OCCUPY48BIT 0x00300000
- #define LV_PIX_EFFECTIVE_PIXEL_SIZE_MASK 0x00FF0000
- #define LV_PIX_EFFECTIVE_PIXEL_SIZE_SHIFT 16

6.11.1 Detailed Description

6.11.2 Macro Definition Documentation

6.11.2.1 #define LV_PIX_COLOR 0x02000000

PixelFormat component: The pixel format is color.

6.11.2.2 #define LV_PIX_COLOR_MASK

Mask for the color flag

6.11.2.3 #define LV_PIX_CUSTOM 0x80000000

PixelFormat component: The pixel format is custom.

6.11.2.4 #define LV_PIX_EFFECTIVE_PIXEL_SIZE_MASK 0x00FF0000

Mask for the pixel size part.

6.11.2.5 #define LV_PIX_EFFECTIVE_PIXEL_SIZE_SHIFT 16

Shift for the pixel size part.

6.11.2.6 #define LV_PIX_MONO 0x01000000

PixelFormat component: The pixel format is monochrome.

6.11.2.7 #define LV_PIX_OCCUPY12BIT 0x000C0000

PixelFormat component: One pixel occupies 12 bits.

6.11.2.8 #define LV_PIX_OCCUPY16BIT 0x00100000

PixelFormat component: One pixel occupies 16 bits.

6.11.2.9 #define LV_PIX_OCCUPY24BIT 0x00180000

PixelFormat component: One pixel occupies 24 bits.

6.11.2.10 #define LV_PIX_OCCUPY32BIT 0x00200000

PixelFormat component: One pixel occupies 32 bits.

6.11.2.11 #define LV_PIX_OCCUPY36BIT 0x00240000

PixelFormat component: One pixel occupies 36 bits.

6.11.2.12 #define LV_PIX_OCCUPY48BIT 0x00300000

PixelFormat component: One pixel occupies 48 bits.

6.11.2.13 #define LV_PIX_OCCUPY8BIT 0x00080000

PixelFormat component: One pixel occupies 8 bits.

6.12 SynView LvLibrary class

Functions

- static UInt32 GetVersion ()
- static LvStatus OpenLibrary ()
- static LvStatus CloseLibrary ()
- static String GetErrorMessage (LvStatus Error)
- static String GetLastErrorMessage ()
- static void Log (String^LogMessage)
- static LvStatus GetLibInfo (LvLibInfo LibInfo, Int32%Info, Int32 Param)
- static LvStatus GetLibInfo (LvLibInfo LibInfo, Int32%Info)
- static LvStatus GetLibInfoStr (LvLibInfo LibInfo, String^%InfoStr)
- static LvStatus GetLibInfoStr (LvLibInfo LibInfo, String^%InfoStr, Int32 Param)
- static LvStatus UpdateSystemList ()
- static LvStatus GetNumberOfSystems (UInt32%NumberOfSystems)
- static LvStatus GetSystemId (UInt32 Index, String^%SystemId)

Variables

- static property Boolean ThrowErrorEnable

6.12.1 Detailed Description

6.12.2 Function Documentation

6.12.2.1 static LvStatus CloseLibrary ( ) [static]

Closes the SynView library. This must be performed before you exit your application. Be sure to close first all dependent modules (System). If you are using SynView in a Windows DLL, avoid calling this in Windows DllMain() function - for proper functionality this function must be called when the application or DLL is still fully functional, which is not the case of PROCESS_DETACH in the DllMain(). If you have called LvOpenLibrary() multiple times, you must balance it by the same number of calls of this function. Only the last call actually does the uninitialization. IMPORTANT: The library must not be opened again once it was already uninitialized.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.2 static String GetErrorMessage (LvStatus *Error*) [static]

Returns a short description of the error. Note that only some of the errors are suitable for direct display to the user, many error values indicate states which are understandable to the programmer, but may not be understandable to the end user.

Parameters

Error	The error code (the return value of most SynView functions).

Returns

The error message.

See also

SynView .Net Class Library Status Definitions.

6.12.2.3 static String GetLastErrorMessage ( ) [static]

Returns more detailed description of the last error, which happened in the thread from which this function was called. As the info is recorded inside SynView for each error, the desription provides more detailed info, including the name of the function, in which the error happened, and possibly more diagnostic info. The difference to LvLibrary::Get↩ ErrorMessage() is that LvLibrary::GetErrorMessage() returns a static string from a numbered table of errors while this function returns additionally info recorded at the time the error happened. If a function returns LVSTATUS_OK, it does not reset this error message (for speed reasons) so the correct approach is to get the error number as the function return value and if this return value is not LVSTATUS_OK, then you can get more info about the error using this function. be sure to call it from the same thread.

Returns

The error message.

See also

SynView .Net Class Library Status Definitions.

6.12.2.4 static LvStatus GetLibInfo (LvLibInfo *LibInfo,* Int32% *Info,* Int32 *Param*) [static]

Gets a general info in form of a 32-bit integer value.

Parameters

LibInfo	One of the LvLibInfo values.
Info	The value is returned in this parameter.
Param	Additional parameter, required by some types of info.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.5 static LvStatus GetLibInfo (LvLibInfo *LibInfo,* Int32% *Info*) [static]

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

6.12.2.6 static LvStatus GetLibInfoStr (LvLibInfo *LibInfo,* String^% *InfoStr*) [static]

Gets a general info in form of a string value.

Parameters

LibInfo	One of the LvLibInfo values.
InfoStr	The string value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.7 static LvStatus GetLibInfoStr (LvLibInfo *LibInfo,* String^% *InfoStr,* Int32 *Param*) [static]

Gets a general info in form of a string value.

Parameters

LibInfo	One of the LvLibInfo values.
InfoStr	The string value is returned in this parameter.
Param	Additional parameter, required by some types of info.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.8 static LvStatus GetNumberOfSystems (UInt32% *NumberOfSystems*) [static]

Returns the number of systems found after the LvUpdateSystemList() call. Typical use of this function is in iterating systems using the LvGetSystemId() function.

Parameters

NumberOf↩ Systems	The number of systems found.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.9 static LvStatus GetSystemId (UInt32 *Index*, String^% *SystemId*) [static]

Returns the string ID of the system at given index. This ID is used in the LvSystem::Open() function for opening the system.

Parameters

Index	Zero-based index of the system, a value >= 0 and < number of systems, returned by the LvGetNumberOfSystems() function.
SystemId	String, where the system ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.10 static UInt32 GetVersion ( ) [static]

Returns SynView version.

Returns

The returned doubleword contains the build version in the low word and the high word is the major version in the upper byte and subversion in the lower byte. For example:

```
UInt32 Version = LvLibrary::GetVersion();
printf("SynView %d.%02d.%03d",
        ((Version >> 24) & 0xFF),
        ((Version >> 16) & 0xFF),
        (Version & 0xFFFF));
```

6.12.2.11 static void Log (String^ *LogMessage*) [static]

Adds a line to the sv.synview.log. The SynView log is a tool for Leutron Vision technical support, but in some cases may be useful to put to the log additional info from your code.

Parameters

LogMessage	String with the message.

6.12.2.12 static LvStatus OpenLibrary ( ) [static]

Opens the SynView library. This must be done before you can use any other SynView function (with the exception of LvLibrary::GetVersion() and LvLibrary::GetErrorMessage()). If you are using SynView in Windows DLL, avoid calling this in Windows DllMain() function - for proper functionality this function must be called when the application or DLL is already fully initialized and there are no restrictions about synchronization (DllMain has such restrictions). If you call this function multiple times, you must balance it by the same number of the LvCloseLibrary() calls. Only the first call will actually do the initialization. IMPORTANT: The library must not be opened again once it was already uninitialized.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.2.13 static LvStatus UpdateSystemList ( ) [static]

Updates the list of systems available. This function must be called before iterating through the systems by the LvGetNumberOfSystems() and LvGetSystemId() functions. The systems are physically represented by GenTL libraries available in the operating systems, this call searches for them in standard locations. See also the description of the lv.synview.ini file in the SynView User's Guide. Note that this function is seldom needed, most applications will work with the default system (see LvSystem::Open() for details).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.12.3 Variable Documentation

6.12.3.1 property Boolean ThrowErrorEnable [static]

Switch to enable/disable the usage of exceptions. If enabled (default), all the methods check the result of the operation and throw LvException if an error status is encountered. If disabled, you should check the function result.

6.13 SynView LvException class

Classes

- class LvException

6.13.1 Detailed Description

6.14 SynView LvSystem class

Functions

- static LvStatus Open (String^ SystemId, LvSystem^%System)
- static LvStatus Close (LvSystem^%System)
- LvStatus UpdateInterfaceList ()
- LvStatus UpdateInterfaceList (UInt32 Timeout)
- LvStatus GetNumberOfInterfaces (UInt32%NumberOfInterfaces)
- LvStatus GetInterfaceId (UInt32 Index, String^%InterfaceId)
- LvStatus FindInterface (LvFindBy FindBy, String^FindStr, String^%InterfaceId)
- LvHSystem GetHandle ()
- LvStatus OpenInterface (String^InterfaceId, LvInterface^%Interface)
- LvStatus CloseInterface (LvInterface^%Interface)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^%Event)
- LvStatus CloseEvent (LvEvent^%Event)

6.14.1 Detailed Description

6.14.2 Function Documentation

6.14.2.1 static LvStatus Close (LvSystem^% *System*) [static]

Deletes the LvSystem class instance. Actually it means freeing the coresponding GenTL library. Be sure you first close all dependent modules (LvInterface, LvEvent etc.). If the System was opened multiple times, it only decreases the reference counter (see the note by the LvSystem::Open()).

Parameters

System	Pointer to LvSystem instance, obtained from the LvSystem::Open() function. The pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.14.2.2 LvStatus CloseEvent (LvEvent^% *Event*)

Deletes the LvEvent class instance. This method is provided just for convenience, it has the same functionality as the LvEvent::Close() static method.

Parameters

Event	Pointer the Event class instance, is assigned nullptr(C++)/null(C#)/Nothing(VB) after the closing is done.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Close().

6.14.2.3 LvStatus CloseInterface (LvInterface^% *Interface*)

Deletes the LvInterface class instance. This method is provided just for convenience, it has the same functionality as the LvInterface::Close() static method. If the Interface was opened multiple times, it only decreases the reference counter (see a note by the LvInterface::Open()). Be sure you first close all dependent modules (LvDevice, LvEvent etc.).

Parameters

Interface	Pointer to the LvInterface instance, obtained from the LvInterface::Open() function. The pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvInterface::Close().

6.14.2.4 LvStatus FindInterface (LvFindBy *FindBy,* String^ *FindStr,* String^% *InterfaceId*)

Finds the interface according specified criteria and returns a string ID of the interface, which is used by the LvInterface::Open() function. This function does not update the interface list - if you need to do so, call the LvSystem::UpdateInterfaceList() function before calling this function.

Parameters

FindBy	Specifies by which criteria to find the interface. Use one of the LvFindBy constants.
FindStr	Specifies the find string, the meaning of which is determined by the FindBy parameter, for example when using the LvFindBy::GevIPAddress, this string should contain the IP address searched for. The searched string is not case sensitive and need not be complete (is searched as a substring).
InterfaceId	String, where the interface ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions. If the Interface is found, the returned status is LVSTATUS_OK.

6.14.2.5 LvHSystem GetHandle ( )

Returns a handle of the System (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.14.2.6 LvStatus GetInterfaceId (UInt32 *Index,* String^% *InterfaceId*)

Returns a string ID of the interface, which is used by the LvInterface::Open() function.

Parameters

Index	Zero-based index of the interface, a value >= 0 and < number of interfaces, returned by the LvSystem::GetNumberOfInterfaces() function.
InterfaceId	String, where the interface ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.14.2.7 LvStatus GetNumberOfInterfaces (UInt32% *NumberOfInterfaces*)

Returns the number of found interfaces, after the LvSystem::UpdateInterfaceList() call.

Parameters

NumberOf↩ Interfaces	Number of interfaces found.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.14.2.8 static LvStatus Open (String^ *SystemId,* LvSystem^% *System*) [static]

Creates the LvSystem class instance. Opening the system actually means loading the corresponding GenT↩ L library. Note that before you can open the System, the LvOpenLibrary() must be called. The same system can be open multiple times (there is a reference counter inside); in such case there must be also the same number of LvSystem::Close() calls used (every open increase the reference count and every close decreases it).

Parameters

SystemId	A string ID of the system. This can be either an empty string - then the default system is opened, or it can be a string obtained from the LvGetSystemId() function.
System	Pointer to the opened LvSystem instance is returned here in case the opening succeeds, nullptr(C++)/null(C#)/Nothing(VB) if fails

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.14.2.9 LvStatus OpenEvent (LvEventType *EventType,* LvEvent^% *Event*)

Creates the LvEvent class instance, owned by the System. This method is provided just for convenience, it has the same functionality as the LvEvent::Open() static method.

Parameters

EventType	One of the LvEventType.

Event	To this parameter the Event class instance is stored.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Open().

6.14.2.10 LvStatus OpenInterface (String^ *InterfaceId*, LvInterface^% *Interface*)

Creates the LvInterface class instance. This method is provided just for convenience, it has the same functionality as the LvInterface::Open() static method. The same Interface can be open multiple times (there is a reference counter inside); in such case there must be also the same number of LvInterface::Close() or LvSystem::InterfaceClose() calls used (every open increase the reference count and every close decreases it).

Parameters

InterfaceId	A string interface ID, obtained by the LvSystem::GetInterfaceId().
Interface	In this parameter the pointer to the LvInterface instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvInterface::Open().

6.14.2.11 LvStatus UpdateInterfaceList ( )

Updates the internal list of available interfaces. You can then iterate through them by LvSystem::GetNumberOf↩Interfaces() and LvSystem::GetInterfaceId().

Returns

The function returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.14.2.12 LvStatus UpdateInterfaceList (UInt32 *Timeout*)

Updates the internal list of available interfaces. You can then iterate through them by LvSystem::GetNumberOf↩Interfaces() and LvSystem::GetInterfaceId().

Parameters

Timeout	Specifies a timeout in ms for searching the interfaces. This applies only to special cases of interfaces, where some delay can happen; common interfaces are detected without any significant delays.

Returns

If the timeout has expired while waiting for the completion, the function returns LVSTATUS_TIMEOUT, otherwise it returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.15 SynView LvInterface class

Functions

- static LvStatus Open (LvSystem^System, String^InterfaceId, LvInterface^%Interface)
- static LvStatus Close (LvInterface^%Interface)
- LvStatus UpdateDeviceList ()
- LvStatus UpdateDeviceList (UInt32 Timeout)
- LvStatus GetNumberOfDevices (UInt32%Devices)
- LvStatus GetDeviceId (UInt32 Index, String^%DeviceId)
- LvStatus FindDevice (LvFindBy FindBy, String^FindStr, String^%DeviceId)
- LvHInterface GetHandle ()
- LvStatus OpenDevice (String^DeviceId, LvDevice^%Device)
- LvStatus OpenDevice (String^DeviceId, LvDevice^%Device, LvDeviceAccess DeviceAccess)
- LvStatus CloseDevice (LvDevice^%Device)
- LvSystem GetParentSystem ()

6.15.1 Detailed Description

6.15.2 Function Documentation

6.15.2.1 static LvStatus Close (LvInterface^% *Interface*) [static]

Deletes the LvInterface class instance. If the Interface was opened multiple times, it only decreases the reference counter (see a note by the LvInterface::Open()). Be sure you first close all dependent modules (LvDevice, LvEvent etc.).

Parameters

Interface	Pointer to the LvInterface instance, obtained from the LvInterface::Open() function. The pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvSystem::CloseInterface().

6.15.2.2 LvStatus CloseDevice (LvDevice^% *Device*)

Deletes the LvDevice class instance. This method is provided just for convenience, it has the same functionality as the LvDevice::Close() static method. Be sure you first close all dependent modules (LvStream, LvEvent etc.).

Parameters

Device	Pointer to the LvDevice instance, obtained from the LvDevice::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::Close().

6.15.2.3 LvStatus FindDevice (LvFindBy *FindBy,* String^ *FindStr,* String^% *DeviceId*)

Finds the device according specified criteria and returns a string ID of the device, which can be used by the LvDevice::Open() function. This function does not update the device list - if you need to do so, call the LvInterface::UpdateDeviceList() function before calling this function.

Parameters

FindBy	Specifies by which criteria to find the interface. Use one of the LvFindBy constants.
FindStr	Specifies the find string, the meaning of which is determined by the FindBy parameter, for example when using the LvFindBy::GevIPAddress, this string should contain the IP address searched for. The searched string is not case sensitive and need not be complete (is searched as a substring).
DeviceId	String, where the device ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions. If the Device is found, the returned status is LVSTATUS_OK.

6.15.2.4 LvStatus GetDeviceId (UInt32 *Index,* String^% *DeviceId*)

Returns a string ID of the device at specified position in the list. Note that this device ID is stable (the same physical device has always the same ID) and it is unique (no other physical device can have the same ID). To hardcode directly the device ID in your application is not recommended, as the application would not be usable, when a deffective device needs to be replaced. The SynView User's Guide discuss the ways, how to solve such maintainability demands.

Parameters

Index	Zero-based index of the device, a value >= 0 and < number of devices, returned by the LvInterface::GetNumberOfDevices() function.
DeviceId	String, where the device ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.15.2.5 LvHInterface GetHandle ( )

Returns a handle of the Interface (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.15.2.6 LvStatus GetNumberOfDevices (UInt32% *Devices*)

Returns the number of devices found by the LvInterface::UpdateDeviceList() function.

Parameters

Devices	Number of devices found.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.15.2.7 LvSystem GetParentSystem ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance.

6.15.2.8 static LvStatus Open (LvSystem^ *System,* String^ *InterfaceId,* LvInterface^% *Interface*) [static]

Creates the LvInterface class instance. The same Interface can be open multiple times (there is a reference counter inside); in such case there must be also the same number of LvInterface::Close() or LvSystem::InterfaceClose() calls used (every open increase the reference count and every close decreases it) .

Parameters

System	A pointer to the LvSystem instance, obtained from the LvSystem::Open() function.
InterfaceId	A string interface ID, obtained by the LvSystem::GetInterfaceId().
Interface	In this parameter the pointer to the LvInterface instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvSystem::OpenInterface().

6.15.2.9 LvStatus OpenDevice (String^ *DeviceId,* LvDevice^% *Device*)

Creates the LvDevice class instance. This method is provided just for convenience, it has the same functionality as the LvDevice::Open() static method. This physically means opening a connection with the device and retrieving a list of device remote features. Always check the success of this function call; the opening may fail for example when you request an exclusive access and the device is already open by some other application.

Parameters

DeviceId	A string ID of the device, obtained by LvInterface::GetDeviceId() function.
Device	In this parameter the pointer to the LvDevice instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::Open().

6.15.2.10 LvStatus OpenDevice (String^ *DeviceId,* LvDevice^% *Device,* LvDeviceAccess *DeviceAccess*)

Creates the LvDevice class instance. This method is provided just for convenience, it has the same functionality as the LvDevice::Open() static method. This physically means opening a connection with the device and retrieving a list of device remote features. Always check the success of this function call; the opening may fail for example when you request an exclusive access and the device is already open by some other application.

Parameters

DeviceId	A string ID of the device, obtained by LvInterface::GetDeviceId() function.
Device	In this parameter the pointer to the LvDevice instance is returned.
DeviceAccess	Desired device access, one of the LvDeviceAccess constants.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::Open().

6.15.2.11 LvStatus UpdateDeviceList ( )

Updates the Device list. The available devices are searched.

Returns

If the timeout has expired while waiting for the completion, the function returns LVSTATUS_TIMEOUT, otherwise it returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.15.2.12 LvStatus UpdateDeviceList (UInt32 *Timeout*)

Updates the Device list. The available devices are searched.

Parameters

Timeout	Specifies a timeout in ms for searching the devices.

Returns

If the timeout has expired while waiting for the completion, the function returns LVSTATUS_TIMEOUT, otherwise it returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16 SynView LvDevice class

Modules

- SynView LvDevice firmware update methods

Functions

- static LvStatus Open (LvInterface^Interface, String^DeviceId, LvDevice^%Device)
- static LvStatus Open (LvInterface^Interface, String^DeviceId, LvDevice^%Device, LvDeviceAccess Device↩ Access)
- static LvStatus Close (LvDevice^%Device)
- LvStatus GetNumberOfStreams (UInt32%NumberOfStreams)
- LvStatus GetStreamId (UInt32 Index, String^%StreamId)
- LvStatus AcquisitionStart ()
- LvStatus AcquisitionStart (UInt32 Options)
- LvStatus AcquisitionStop ()
- LvStatus AcquisitionStop (UInt32 Options)
- LvStatus AcquisitionAbort ()
- LvStatus AcquisitionAbort (UInt32 Options)
- LvStatus AcquisitionArm ()
- LvStatus AcquisitionArm (UInt32 Options)
- LvStatus SaveSettings (String^Id, String^FileName, UInt32 Options)
- LvStatus LoadSettings (String^Id, String^FileName, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< Byte >^ByteArray, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< Byte >^ByteArray, SizeT Size, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, array< UInt16 >^UInt16Array, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< UInt16 >^UInt16Array, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< Byte >^ByteArray, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< Byte >^ByteArray, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, array< UInt16 >^UInt16Array, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< UInt16 >^UInt16Array, SizeT Size, UInt32 Options)
- LvStatus OpenStream (String^StreamId, LvStream^%Stream)
- LvStatus CloseStream (LvStream^%Stream)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^%Event)
- LvStatus CloseEvent (LvEvent^%Event)
- LvHDevice GetHandle ()

6.16.1 Detailed Description

6.16.2 Function Documentation

6.16.2.1 LvStatus AcquisitionAbort ( )

Aborts athe acquisition immediately, without completing the current Frame or waiting on a trigger.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.2 LvStatus AcquisitionAbort (UInt32 *Options*)

Aborts athe acquisition immediately, without completing the current Frame or waiting on a trigger.

Parameters

Options	Reserved for future use, must be 0 or ommitted.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.3 LvStatus AcquisitionArm ( )

Prepares the device for acquisition, so that the acquisition using the LvDevice::AcquisitionStart() function then can start fast. If it is not called before LvDevice::AcquisitionStart(), it is called automatically inside the LvDevice::AcquisitionStart().

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.4 LvStatus AcquisitionArm (UInt32 *Options*)

Prepares the device for acquisition, so that the acquisition using the LvDevice::AcquisitionStart() function then can start fast. If it is not called before LvDevice::AcquisitionStart(), it is called automatically inside the LvDevice::AcquisitionStart().

Parameters

Options	Reserved for future use, must be 0 or ommitted.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.5 LvStatus AcquisitionStart ( )

Starts the acquisition. This function includes more than just calling the remote AcquisitionStart command on the device - it checks the size of the buffers, prepares the streams for the start, locks GenTL params and then starts the acquisition on the device itself. Always check the success of this function call.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.6 LvStatus AcquisitionStart (UInt32 *Options*)

Starts the acquisition. This function includes more than just calling the remote AcquisitionStart command on the device - it checks the size of the buffers, prepares the streams for the start, locks GenTL params and then starts the acquisition on the device itself. Always check the success of this function call.

Parameters

Options	Reserved for future use, must be 0 or ommitted.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.7 LvStatus AcquisitionStop ( )

Stops the acquisition.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.8 LvStatus AcquisitionStop (UInt32 *Options*)

Stops the acquisition.

Parameters

Options	Reserved for future use, must be 0 or ommitted.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.9 static LvStatus Close (LvDevice^% *Device*) [static]

Deletes the LvDevice class instance. Be sure you first close all dependent modules (LvStream, LvEvent etc.).

Parameters

Device	Pointer to the LvDevice instance, obtained from the LvDevice::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvInterface::CloseDevice().

6.16.2.10 LvStatus CloseEvent (LvEvent^% *Event*)

Deletes the LvEvent class instance. This method is provided just for convenience, it has the same functionality as the LvEvent::Close() static method.

Parameters

Event	A pointer to the LvEvent class instance, obtained from the LvEvent::Open() or LvDevice::OpenEvent() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Close().

6.16.2.11 LvStatus CloseStream (LvStream^% *Stream*)

Deletes the LvStream class instance. This method is provided just for convenience, it has the same functionality as the LvStream::Close() static method. Be sure you first close all dependent modules (LvBuffers, LvEvent, LvRenderer etc.).

Parameters

Stream	Pointer to the LvStream instance, obtained from the LvStream::Open() or LvDevice::OpenStream() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::Close().

6.16.2.12 LvHDevice GetHandle ( )

Returns a handle of the Device (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.16.2.13 LvStatus GetNumberOfStreams (UInt32% *NumberOfStreams*)

Returns the number of available stream types for this device. You can then iterate the streams by the LvDevice::GetStreamId() function.

Parameters

NumberOf↩ Streams	The number of streams is returned here.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.14 LvStatus GetStreamId (UInt32 *Index*, String^% *StreamId*)

Returns a string Stream ID, needed for opening the stream.

Parameters

Index	Zero-based index of the stream type, a value >= 0 and < number of streams, returned by the LvDevice::GetNumberOfStreams() function.
StreamId	String, where the stream ID is placed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.15 LvStatus LoadSettings (String^ *Id*, String^ *FileName*, UInt32 *Options*)

Loads the device settings from a file. In the Options can be specified which parts of the device configuration are to be loaded - the Remote, Local and/or GenTL part. Note that there are several factors, which can break the compatibility of the settings file with the current device:

- When the current device is of different vendor/model, the settings file is most probably not compatible.
- When the current device is of the same vendor/model, but uses a different firmware version - this could mean that some remote device features are not present or behave differently.
- When the XML version of the Local and GenTL features changes - again this could mean that some features are not present or behave differently. For this reason this function checks the versions and if not the same, it returns either the LVSTATUS_SETTINGS_INCOMPATIBLE_MODEL or LVSTATUS_SETTINGS↩ _INCOMPATIBLE_VERSION error states. As the difference in versions might not necessarily mean a real incompability, you can use the LvSaveFlag::IgnoreVersion and LvSaveFlag::IgnoreModel flags in the Options parameter in order to force this function to try to load the settings even if the possible incompatibility is found.

Parameters

Id	A string ID enabling to protect the file. If you specify a non-empty ID in LvDevice::Save↩ Settings(), you must use the same ID in LvDevice::LoadSettings(), otherwise the settings are not loaded.
FileName	The file specification, where the configuration is stored. It is a text file.
Options	One or or-ed combination of LvSaveFlag.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.16 static LvStatus Open (LvInterface^ *Interface,* String^ *DeviceId,* LvDevice^% *Device*) [static]

Creates the LvDevice class instance.

This physically means opening a connection with the device and retrieving a list of device remote features. Always check the success of this function call; the opening may fail for example when you request an exclusive access and the device is already open by some other application.

Parameters

Interface	A pointer to the LvInterface instance, obtained from the LvInterface::Open() function.
DeviceId	A string ID of the device, obtained by LvInterface::GetDeviceId() function.
Device	In this parameter the pointer to the LvDevice instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvInterface::OpenDevice().

6.16.2.17 static LvStatus Open (LvInterface^ *Interface,* String^ *DeviceId,* LvDevice^% *Device,* LvDeviceAccess *DeviceAccess*) [static]

Creates the LvDevice class instance.

This physically means opening a connection with the device and retrieving a list of device remote features. Always check the success of this function call; the opening may fail for example when you request an exclusive access and the device is already open by some other application.

Parameters

Interface	A pointer to the LvInterface instance, obtained from the LvInterface::Open() function.
DeviceId	A string ID of the device, obtained by LvInterface::GetDeviceId() function.
Device	In this parameter the pointer to the LvDevice instance is returned.
DeviceAccess	Desired device access, one of the LvDeviceAccess constants.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvInterface::OpenDevice().

6.16.2.18 LvStatus OpenEvent (LvEventType *EventType,* LvEvent^% *Event*)

Creates the LvEvent class instance for specified owner module. This method is provided just for convenience, it has the same functionality as the LvEvent::Open() static method.

Parameters

EventType	One of the LvEventType.
Event	To this parameter the pointer LvEvent instance is stored.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Open().

6.16.2.19 LvStatus OpenStream (String^ *StreamId,* LvStream^% *Stream*)

Creates the LvStream class instance associated with the device. This method is provided just for convenience, it has the same functionality as the LvStream::Open() static method.

Parameters

StreamId	A string ID of the stream, obtained from LvDevice::GetStreamId(). If an empty string is used, the first found stream is opened. This is usually the image data stream.
Stream	In this parameter the pointer to the LvStream instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::Open().

6.16.2.20 LvStatus SaveSettings (String^ *Id,* String^ *FileName,* UInt32 *Options*)

Saves the device settings to a file. In the Options can be specified which parts of the device configuration are to be saved - the Remote, Local and/or GenTL part. See alo notes by LvDevice::LoadSettings().

Parameters

Id	A string ID enabling to protect the file. If you specify a non-empty ID in LvDevice::SaveSettings(), you must use the same ID in LvDevice::LoadSettings(), otherwise the settings are not loaded.
FileName	The file specification, to which the configuration is stored. It is a text file.
Options	One or or-ed combination of LvSaveFlag .

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.21 LvStatus UniGetLut (LvLUTSelector *Selector,* IntPtr *pLUT,* SizeT *Size,* UInt32 *Options*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
pLUT	Pointer to the lookup table (passed as unmanaged pointer).
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT • 2048 for 10-bit LUT • 8192 for 12-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.22 LvStatus UniGetLut (LvLUTSelector *Selector,* IntPtr *pLUT,* SizeT *Size*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
pLUT	Pointer to the lookup table (passed as unmanaged pointer).
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT • 2048 for 10-bit LUT • 8192 for 12-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.23 LvStatus UniGetLut (LvLUTSelector *Selector,* array< Byte >^ *ByteArray,* SizeT *Size*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
ByteArray	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.24 LvStatus UniGetLut (LvLUTSelector *Selector*, array< Byte >^ *ByteArray*, SizeT *Size*, UInt32 *Options*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
ByteArray	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.25 LvStatus UniGetLut (LvLUTSelector *Selector*, array< UInt16 >^ *UInt16Array*, SizeT *Size*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
UInt16Array	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 1024 for 10-bit LUT • 4096 for 12-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.26 LvStatus UniGetLut (LvLUTSelector *Selector*, array< UInt16 >^ *UInt16Array*, SizeT *Size*, UInt32 *Options*)

Gets the lookup table. See LvDevice::UniSetLut() for details. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format. So you can for example read the 12-bit LUT to 8-bit LUT array.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
UInt16Array	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 1024 for 10-bit LUT • 4096 for 12-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.27 LvStatus UniSetLut (LvLUTSelector *Selector,* IntPtr *pLUT,* SizeT *Size,* UInt32 *Options*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
pLUT	Pointer to the lookup table (passed as unmanaged pointer).
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT • 2048 for 10-bit LUT • 8192 for 12-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.28 LvStatus UniSetLut (LvLUTSelector *Selector,* IntPtr *pLUT,* SizeT *Size*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.

pLUT	Pointer to the lookup table (passed as unmanaged pointer).
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT • 2048 for 10-bit LUT • 8192 for 12-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.29 LvStatus UniSetLut (LvLUTSelector *Selector,* array< Byte >^ *ByteArray,* SizeT *Size*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
ByteArray	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT • 2048 for 10-bit LUT • 8192 for 12-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.30 LvStatus UniSetLut (LvLUTSelector *Selector,* array< Byte >^ *ByteArray,* SizeT *Size,* UInt32 *Options*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
ByteArray	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 256 for 8-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.31 LvStatus UniSetLut (LvLUTSelector *Selector,* array< UInt16 >^ *UInt16Array,* SizeT *Size*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
UInt16Array	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 1024 for 10-bit LUT • 4096 for 12-bit LUT

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.16.2.32 LvStatus UniSetLut (LvLUTSelector *Selector,* array< UInt16 >^ *UInt16Array,* SizeT *Size,* UInt32 *Options*)

Sets the lookup table. If the hardware lookup table is available and enabled, it is used, otherwise a software lookup table is set. This function belongs to a set of functions, which unify the functionality of devices with real-time processing embedded in hardware (RTF) and devices without real-time processing, for which the processing is made by software. The LUT is automatically recalculated to appropriate type, if you use different LUT bit depth than is the actually used for the current pixel format.

Parameters

Selector	Lookup table selector, see LvLUTSelector.
UInt16Array	The LUT as a managed array.
Size	Size of the lookup table. The only valid values are • 1024 for 10-bit LUT • 4096 for 12-bit LUT
Options	Reserved for future use, must be 0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.17 SynView LvDevice firmware update methods

Functions

- LvStatus FwGetFilePattern (UInt32 Which, String^%FilePattern)
- LvStatus FwLoad (UInt32 Which, String^FilePath)
- LvStatus FwGetLoadStatus (UInt32 Which, UInt32%CurrentByteCount, Boolean%IsLoading)

6.17.1 Detailed Description

6.17.2 Function Documentation

6.17.2.1 LvStatus FwGetFilePattern (UInt32 *Which,* String^% *FilePattern*)

Returns the file name mask (with wildcard characters), for searching the file with the appropriate firmware update. The files with the FW update have in their names coded the hardware IDs, so using this mask (for example in a filter in a file open dialog box) assures the file appropriate for this device is used.

Parameters

Which	An ID specific for a hardware. Discussed in the SynView User's Guide.
FilePattern	In this parameter the file pattern is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.17.2.2 LvStatus FwGetLoadStatus (UInt32 *Which,* UInt32% *CurrentByteCount,* Boolean% *IsLoading*)

Returns the byte count and whether the loading is still in progress.

Parameters

Which	An ID specific for a hardware. Discussed in the SynView User's Guide.
CurrentByteCount	Returns number of bytes transferred so far.
IsLoading	Returns true if the loading is still in progress.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.17.2.3 LvStatus FwLoad (UInt32 *Which,* String^ *FilePath*)

Loads the firmware from a file to the hardware. It can be very long process (taking minutes) and this functions blocks the thread druting this process. It is recommended to check the load status from another thread using the LvFwGetLoadStatus() function.

Parameters

Which	An ID specific for a hardware. Discussed in the SynView User's Guide.
FilePath	File specification, with full path.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18 SynView LvStream class

Functions

- LvInterface GetParentInterface ()
- static LvStatus Open (LvDevice^Device, String^StreamId, LvStream^%Stream)
- static LvStatus Close (LvStream^%Stream)
- LvStatus GetBufferAt (UInt32 BufferIndex, LvBuffer^%Buffer)
- LvStatus FlushQueue (LvQueueOperation QueueOperation)
- LvStatus Start ()
- LvStatus Start (LvStreamStartFlags StartFlags)
- LvStatus Start (LvStreamStartFlags StartFlags, UInt32 ImagesToAcquire)
- LvStatus Stop ()
- LvStatus Stop (LvStreamStopFlags StopFlags)
- LvHStream GetHandle ()
- LvStatus OpenBuffer (IntPtr pDataPointer, SizeT DataSize, IntPtr pUserPointer, UInt32 Options, LvBuffer^%Buffer)
- LvStatus CloseBuffer (LvBuffer^%Buffer)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^%Event)
- LvStatus CloseEvent (LvEvent^%Event)
- LvStatus OpenRenderer (LvRenderer^%Renderer)
- LvStatus CloseRenderer (LvRenderer^%Renderer)
- LvDevice GetParentDevice ()

6.18.1 Detailed Description

6.18.2 Function Documentation

6.18.2.1 static LvStatus Close (LvStream^% *Stream*) [static]

Deletes the LvStream class instance. Be sure you first close all dependent modules (LvBuffers, LvEvent, LvRenderer etc.).

Parameters

Stream	Pointer to the LvStream instance, obtained from the LvStream::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::CloseStream().

6.18.2.2 LvStatus CloseBuffer (LvBuffer^% *Buffer*)

Deletes the LvBuffer class instance. On the GenTL level it corresponds to the DSRevokeBuffer() function. This method is provided just for convenience, it has the same functionality as the LvBuffer::Close() static method.

Parameters

Buffer	A pointer to the LvBuffer instance, obtained from the LvBuffer::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvBuffer::Close().

6.18.2.3 LvStatus CloseEvent (LvEvent^% *Event*)

Deletes the LvEvent class instance. This method is provided just for convenience, it has the same functionality as the LvEvent::Close() static method.

Parameters

Event	Pointer the Event class instance, is assigned nullptr(C++)/null(C#)/Nothing(VB) after the closing is done.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Close().

6.18.2.4 LvStatus CloseRenderer (LvRenderer^% *Renderer*)

Deletes the LvRenderer class instance. This method is provided just for convenience, it has the same functionality as the LvRenderer::Close() static method.

Parameters

Renderer	A pointer to the LvRenderer instance, obtained from the LvRenderer::Open() function. This pointer is set to nullptr(C++)/null(C#)/Nothing(VB) after close.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvRenderer::Close().

6.18.2.5 LvStatus FlushQueue (LvQueueOperation *QueueOperation*)

Moves the buffers according to the LvQueueOperation specified.

Parameters

QueueOperation	One of the LvQueueOperation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.6 LvStatus GetBufferAt (UInt32 *BufferIndex,* LvBuffer^% *Buffer*)

Returns the buffer instance at given index.

Parameters

BufferIndex	Zero-based index.
Buffer	In this parameter the pointer to LvBuffer instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.7 LvHStream GetHandle ( )

Returns a handle of the Stream (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.18.2.8 LvDevice GetParentDevice ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance.

6.18.2.9 LvInterface GetParentInterface ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance.

6.18.2.10 static LvStatus Open (LvDevice^ *Device,* String^ *StreamId,* LvStream^% *Stream*) [static]

Creates the LvStream class instance, associated with the device.

Parameters

Device	A pointer to the LvDevice instance, obtained from the LvDevice::Open() function.
StreamId	A string ID of the stream, obtained from LvDevice::GetStreamId(). If an empty string is used, the first found stream is opened. This is usually the image data stream.
Stream	In this parameter the pointer to the LvStream instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::OpenStream().

6.18.2.11 LvStatus OpenBuffer (IntPtr *pDataPointer,* SizeT *DataSize,* IntPtr *pUserPointer,* UInt32 *Options,* LvBuffer^% *Buffer*)

Creates the LvBuffer class instance. On the GenTL level it corresponds to DSAnnounceBuffer() or DSAllocAnd↩ AnnounceBuffer(). This method is provided just for convenience, it has the same functionality as the LvBuffer::↩ Open() static method.

Parameters

pDataPointer	Pointer to image data buffer. This can be supplied by the application (in such case the DataSize must be set to the actual size of the buffer), or can be left nullptr(C++)/null(C#)/↩ Nothing(VB) - in such case the buffer is allocated by SynView.
DataSize	Size of the buffer supplied, or 0 if the pDataPointer is nullptr(C++)/null(C#)/Nothing(VB).
pUserPointer	A user pointer, which is then passed back in the LvEventCallbackNewBufferFunct(). It enables the application to reference some own data structure associated with the buffer.
Options	Currently unused, must be 0.
Buffer	To this parameter the pointer to the LvBuffer instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvBuffer::Open().

6.18.2.12 LvStatus OpenEvent (LvEventType *EventType,* LvEvent^% *Event*)

Creates the LvEvent class instance, owned by the Stream. This method is provided just for convenience, it has the same functionality as the LvEvent::Open() static method.

Parameters

EventType	One of the LvEventType.
Event	To this parameter the Event class instance is stored.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvEvent::Open().

6.18.2.13 LvStatus OpenRenderer (LvRenderer^% *Renderer*)

Creates the LvRenderer class instance for image display. The renderer attepts to load the lv.synview.display library. In case of SynView installation in an operating system without possibility to graphically display (for example Linux without XWindows), the load of this library fails and the calls to Renderer functions will return errors. This method is provided just for convenience, it has the same functionality as the LvRenderer::Open() static method.

Parameters

Renderer	In this parameter the pointer to the LvRenderer is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvRenderer::Open().

6.18.2.14 LvStatus Start ( )

Starts the stream. This function need not be used on the image stream, where it is called automatically in the LvDevice::AcquisitionStart() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.15 LvStatus Start (LvStreamStartFlags *StartFlags*)

Starts the stream. This function need not be used on the image stream, where it is called automatically in the LvDevice::AcquisitionStart() function.

Parameters

StartFlags	One of the GroupSynView_StreamStartFlags.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.16 LvStatus Start (LvStreamStartFlags *StartFlags,* UInt32 *ImagesToAcquire*)

Starts the stream. This function need not be used on the image stream, where it is called automatically in the LvDevice::AcquisitionStart() function.

Parameters

StartFlags	One of the GroupSynView_StreamStartFlags.
ImagesTo↩ Acquire	Number of images to acquire.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.17 LvStatus Stop ( )

Stops the stream. This function need not be used on the image stream, where it is called automatically in the LvDevice::AcquisitionStop() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.18.2.18 LvStatus Stop (LvStreamStopFlags *StopFlags*)

Stops the stream. This function need not be used on the image stream, where it is called automatically in the LvDevice::AcquisitionStop() function.

Parameters

StopFlags	One of the GroupSynView_StreamStopFlags.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19 SynView LvBuffer class

Functions

- static LvStatus Open (LvStream^Stream, IntPtr pDataPointer, SizeT DataSize, IntPtr pUserPointer, UInt32 Options, LvBuffer^%Buffer)
- static LvStatus Close (LvBuffer^%Buffer)
- LvStatus AttachProcessBuffer (IntPtr pDataPointer, SizeT DataSize)
- LvStatus Queue ()
- LvStatus ParseChunkData ()
- LvStatus ParseChunkData (Boolean UpdateLayout)
- LvStatus SaveImageToBmpFile (String^FileName)
- LvStatus SaveImageToJpgFile (String^FileName, UInt32 Quality)
- LvStatus SaveImageToTifFile (String^FileName)
- LvStatus SaveImageToTifFile (String^FileName, UInt32 Options)
- LvStatus GetLvipImage (LvipImage^%Image)
- LvStatus GetLastPaintRect (Int32%X, Int32%Y, Int32%Width, Int32%Height)
- LvStatus UniCalculateWhiteBalance ()
- LvHBuffer GetHandle ()
- IntPtr GetUserPtr ()
- LvStream GetParentStream ()

6.19.1 Detailed Description

6.19.2 Function Documentation

6.19.2.1 LvStatus AttachProcessBuffer (IntPtr *pDataPointer,* SizeT *DataSize*)

Attaches a process buffer to a buffer. The process buffer may be needed for software processing, for example Bayer decoding, if the device hardware is not capable of it. The process buffer can be either supplied by the application by this function, or allocated automatically by SynView, upon need.

Parameters

pDataPointer	Pointer to the supplied buffer.
DataSize	Size of the buffer.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.2 static LvStatus Close (LvBuffer^% *Buffer*) [static]

Deletes the LvBuffer class instance. On the GenTL level it corresponds to the DSRevokeBuffer() function.

Parameters

Buffer	A pointer to the LvBuffer instance, obtained from the LvBuffer::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::CloseBuffer().

6.19.2.3 LvHBuffer GetHandle ( )

Returns a handle of the Buffer (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.19.2.4 LvStatus GetLastPaintRect (Int32% *X*, Int32% *Y*, Int32% *Width*, Int32% *Height*)

Returns the rectangle to which the buffer was last painted. This is useful namely in case you have a tile mode and want to identify the buffer according a mouse click location. If the buffer was not yet painted by the renderer, the returned values are 0.

Parameters

X	X offset in pixels.
Y	Y offset in pixels.
Width	Width in pixels.
Height	Height in pixels.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.5 LvStatus GetLvipImage (LvipImage^% *Image*)

Fills the LvipImage class instance for the image in the buffer. This simplifies a direct use of the SynView Image Processing .Net Class Library. If the image is processed, the image info points to the processed image, othewise it points to the original image.

Parameters

Image	Pointer to the LvipImage class, to which are the image parameters stored.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.6 LvStream GetParentStream ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance.

6.19.2.7 IntPtr GetUserPtr ( )

Returns the user pointer associated with the buffer.

Note that the feature of the same name is in the .Net Class Library used for internal purposes and must not be set by the user.

Returns

The user pointer.

6.19.2.8 static LvStatus Open (LvStream^ *Stream*, IntPtr *pDataPointer*, SizeT *DataSize*, IntPtr *pUserPointer*, UInt32 *Options*, LvBuffer^% *Buffer*) [static]

Creates the LvBuffer class instance. On the GenTL level it corresponds to DSAnnounceBuffer() or DSAllocAnd↩ AnnounceBuffer().

Parameters

Stream	A pointer to the LvStream instance, obtained from the LvStream::Open() function.
pDataPointer	Pointer to image data buffer. This can be supplied by the application (in such case the DataSize must be set to the actual size of the buffer), or can be left nullptr(C++)/null(C#)/↩ Nothing(VB) - in such case the buffer is allocated by SynView.
DataSize	Size of the buffer supplied, or 0 if the pDataPointer is nullptr(C++)/null(C#)/Nothing(VB).
pUserPointer	A user pointer, which is then passed back in the LvEventCallbackNewBufferFunct(). It enables the application to reference some own data structure associated with the buffer.
Options	Currently unused, must be 0.
Buffer	To this parameter the pointer to the LvBuffer instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::OpenBuffer().

6.19.2.9 LvStatus ParseChunkData ( )

Parses the chunk data of the image. The chunk data are then accessible as device remote features (for example LvDevice::ChunkTimestamp).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.10 LvStatus ParseChunkData (Boolean *UpdateLayout*)

Parses the chunk data of the image. The chunk data are then accessible as device remote features (for example LvDevice::ChunkTimestamp).

Parameters

UpdateLayout	If set to true, the layout of chunk data is decoded. If set to false, the data are only read from already decoded layout - this is faster. Usually, the layout of the chunk data is constant, so it needs to be decoded only at first call of this function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.11 LvStatus Queue ( )

Puts the buffer to the input buffer pool. This is an important part of the image handling loop: after the buffer with the acquired image is passed to the application, the application must return it to the input buffer pool by this function after processing.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.12 LvStatus SaveImageToBmpFile (String^ *FileName*)

Saves the image to a file in Windows BMP format. If the image is in the pixel format not compatible with the BMP format, it is automatically converted.

Parameters

FileName	The file name. Be sure to specify it with the full path.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.13 LvStatus SaveImageToJpgFile (String^ *FileName,* UInt32 *Quality*)

Saves the image to a file in JPEG format. If the image is in the pixel format not compatible with the JPEG format, it is automatically converted.

Parameters

FileName	The file name. Be sure to specify it with the full path.
Quality	The quality factor in range from from 1 to 100. The higher is the factor, the higher is the quality and lower the compression.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.14 LvStatus SaveImageToTifFile (String^ *FileName*)

Saves the image to a file in the TIFF format. If the image is in the pixel format not compatible with the TIF format, it is automatically converted.

Parameters

FileName	The file name. Be sure to specify it with the full path.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.15 LvStatus SaveImageToTifFile (String^ *FileName*, UInt32 *Options*)

Saves the image to a file in the TIFF format. If the image is in the pixel format not compatible with the TIF format, it is automatically converted.

Parameters

FileName	The file name. Be sure to specify it with the full path.
Options	Options for saved pixel format. The LvipOption::TiffConvertTo16Bit flag can be used there.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.19.2.16 LvStatus UniCalculateWhiteBalance ( )

Calculates white balance factors from the current image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20 SynView LvEvent class

Functions

- static LvStatus Open (LvSystem^System, LvEventType EventType, LvEvent^%Event)
- static LvStatus Open (LvDevice^Device, LvEventType EventType, LvEvent^%Event)
- static LvStatus Open (LvStream^Stream, LvEventType EventType, LvEvent^%Event)
- static LvStatus Close (LvEvent^%Event)
- LvStatus Kill ()
- LvStatus Flush ()
- LvStatus WaitAndGetData (IntPtr pBuffer, SizeT%Size, UInt32 Timeout)
- LvStatus WaitAndGetData (IntPtr pBuffer, SizeT%Size)
- LvStatus WaitAndGetNewBuffer (LvBuffer^%Buffer, UInt32 Timeout)
- LvStatus WaitAndGetNewBuffer (LvBuffer^%Buffer)
- LvStatus GetDataInfo (IntPtr pInBuffer, SizeT InSize, LvEventDataInfo EventDataInfo, IntPtr pBuffer, Size↩ T%Size, LvInfoDataType%InfoDataType, Int32 Param)
- LvStatus PutData (IntPtr pBuffer, SizeT Size)
- LvStatus SetCallback (Boolean Set, IntPtr pUserParam)
- LvStatus SetCallbackNewBuffer (Boolean Set, IntPtr pUserParam)
- LvStatus StartThread ()
- LvStatus StopThread ()
- LvHEvent GetHandle ()
- LvSystem GetParentSystem ()
- LvDevice GetParentDevice ()
- LvStream GetParentStream ()

6.20.1 Detailed Description

6.20.2 Function Documentation

6.20.2.1 static LvStatus Close (LvEvent^% *Event*) [static]

Deletes the LvEvent class instance.

Parameters

Event	A pointer to the LvEvent instance, obtained from the LvEvent::Open() function. This pointer is assigned nullptr(C++)/null(C#)/Nothing(VB) after the operation.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvSystem::CloseEvent(), LvDevice::CloseEvent(), LvStream::CloseEvent().

6.20.2.2 LvStatus Flush ( )

Discards all buffers in the output buffer queue (waiting to be delivered to the application).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.3 LvStatus GetDataInfo (IntPtr *pInBuffer,* SizeT *InSize,* LvEventDataInfo *EventDataInfo,* IntPtr *pBuffer,* SizeT% *Size,* LvInfoDataType% *InfoDataType,* Int32 *Param*)

Enables to parse the buffer from LvEvent::WaitAndGetData.

Parameters

pInBuffer	Pointer to a buffer containing event data. This value must not be nullptr(C++)/null(C#)/Nothing(VB).
InSize	Size of the provided pInBuffer in bytes.
EventDataInfo	One of the LvEventDataInfo.
pBuffer	Pointer to a user allocated buffer to receive the requested information. If this parameter is nullptr(C++)/null(C#)/Nothing(VB), pSize will contain the minimal size of pBuffer in bytes. If the pType is a string, the size includes the terminating 0.
Size	Size of the buffer is returned in this parameter.
InfoDataType	One of the LvInfoDataType.
Param	Additional parameter, if used, its role is explained by the LvEventDataInfo.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.4 LvHEvent GetHandle ( )

Returns a handle of the Event (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.20.2.5 LvDevice GetParentDevice ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance. If the class instance was not opened for LvDevice, returns nullptr.

6.20.2.6 LvStream GetParentStream ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance. If the class instance was not opened for LvStream, returns nullptr.

6.20.2.7 LvSystem GetParentSystem ( )

Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance. If the class instance was not opened for LvSystem, returns nullptr.

6.20.2.8 LvStatus Kill ( )

Terminates a single wait in the LvEvent::WaitAndGetData() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.9 static LvStatus Open (LvSystem^ *System*, LvEventType *EventType*, LvEvent^% *Event*) [static]

Creates the LvEvent class instance for specified LvSystem module.

Parameters

System	Pointer to LvSystem class instance.
EventType	One of the LvEventType.
Event	In this parameter a pointer to LvEvent instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvSystem::OpenEvent().

6.20.2.10 static LvStatus Open (LvDevice^ *Device*, LvEventType *EventType*, LvEvent^% *Event*) [static]

Creates the LvEvent class instance for specified LvDevice module.

Parameters

Device	Pointer to LvDevice class instance.
EventType	One of the LvEventType.
Event	In this parameter a pointer to LvEvent instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvDevice::OpenEvent().

6.20.2.11 static LvStatus Open (LvStream^ *Stream*, LvEventType *EventType*, LvEvent^% *Event*) [static]

Creates the LvEvent class instance for specified LvStream module.

Parameters

Stream	Pointer to LvStream class instance.
EventType	One of the LvEventType.
Event	In this parameter a pointer to LvEvent instance is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::OpenEvent().

6.20.2.12 LvStatus PutData (IntPtr *pBuffer,* SizeT *Size*)

Puts a new event to Event ouptut queue. This function can be used only for user-defined events.

Parameters

pBuffer	Pointer to event data.
Size	Size of the event data.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.13 LvStatus SetCallback (Boolean *Set,* IntPtr *pUserParam*)

Activates or deactivates a callback function for the event thread. The callback function itself is implemented internally and is converted to an event, see OnEvent, so in this function you only specify if the callback should be registered or unregistered.

Parameters

Set	If set to true, the callback is internaly registered and converted into the event. If set to false, the callback is internally unregistered.
pUserParam	User parameter, which will be passed to each callback call.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.14 LvStatus SetCallbackNewBuffer (Boolean *Set,* IntPtr *pUserParam*)

Activates or deactivates a callback function for the thread of the Event of the LvEventType::NewBuffer. The callback function itself is implemented internally and is converted to an event, see LvEvent::OnEventNewBuffer, so in this function you only specify if the callback should be registered or unregistered. Once the application activates this callback, it becomes responsible for returning the image buffers to the input buffer pool.

Parameters

Set	If set to true, the callback is internaly registered and converted into the event. If set to false, the callback is internally unregistered.
pUserParam	User parameter, which will be passed to each callback call.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.15 LvStatus StartThread ( )

Starts an internal thread, which waits for events and passes them to specified callback function. When the thread is started, the application must no longer call the LvEvent::WaitAndGetData() or LvEvent::WaitAndGetNewBufer() functions - this is called internally in the thread and upon return from this function a callback function is called.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.16 LvStatus StopThread ( )

Stops the event internal thread.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.17 LvStatus WaitAndGetData (IntPtr *pBuffer*, SizeT% *Size*, UInt32 *Timeout*)

Waits for the event and gets its data in one atomic operation. Use this function only for events other than LvEvent↩ Type::NewBuffer, for the the LvEventType::NewBuffer event type use the LvEvent::WaitAndGetNewBuffer() function instead. Do not use this function if you use the callback - see LvEvent::SetCallback() or LvEvent::SetCallbackNew↩ Buffer().

Parameters

pBuffer	Pointer to a user allocated buffer to receive the event data. The buffer can be parsed by the LvEvent::GetDataInfo() function.
Size	Size of the buffer must be specified in this parameter and after the function returns, the actual size is returned in this parameter.
Timeout	The wait timeout in milliseconds. The value 0xFFFFFFFF is considered as infinite. Note that you can also kill waiting from another thread using the LvEvent::Kill() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.18 LvStatus WaitAndGetData (IntPtr *pBuffer,* SizeT% *Size*)

Waits for the event and gets its data in one atomic operation. Use this function only for events other than LvEvent↩ Type::NewBuffer, for the the LvEventType::NewBuffer event type use the LvEvent::WaitAndGetNewBuffer() function instead. Do not use this function if you use the callback - see LvEvent::SetCallback() or LvEvent::SetCallbackNew↩ Buffer().

Parameters

pBuffer	Pointer to a user allocated buffer to receive the event data. The buffer can be parsed by the LvEvent::GetDataInfo() function.
Size	Size of the buffer must be specified in this parameter and after the function returns, the actual size is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.19 LvStatus WaitAndGetNewBuffer (LvBuffer^% *Buffer,* UInt32 *Timeout*)

Waits for the event and gets its data in one atomic operation. Use this function only for events of the LvEventType::NewBuffer type. Do not use this function if you use the callback - see LvEvent::SetCallback() or LvEvent::SetCallbackNewBuffer().

Parameters

Buffer	The pointer to the received LvBuffer instance is returned in this parameter.
Timeout	The wait timeout in milliseconds. The value 0xFFFFFFFF is considered as infinite. Note that you can also kill waiting from another thread using the LvEvent::Kill() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.20.2.20 LvStatus WaitAndGetNewBuffer (LvBuffer^% *Buffer*)

Waits for the event and gets its data in one atomic operation. Use this function only for events of the LvEventType::NewBuffer type. Do not use this function if you use the callback - see LvEvent::SetCallback() or LvEvent::SetCallbackNewBuffer().

Parameters

Buffer	The pointer to the received LvBuffer instance is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.21 SynView LvRenderer class

Functions

- static LvStatus Open (LvStream^Stream, LvRenderer^%Renderer)
- static LvStatus Close (LvRenderer^%Renderer)
- LvStatus SetWindow (IntPtr hWnd)
- LvStatus CanDisplayImage (LvBuffer^Buffer, UInt32 RenderFlags)
- LvStatus CanDisplayImage (LvBuffer^Buffer)
- LvStatus DisplayImage (LvBuffer^Buffer, UInt32 RenderFlags)
- LvStatus DisplayImage (LvBuffer^Buffer)
- LvStatus Repaint (UInt32 RenderFlags)
- LvStatus Repaint ()
- LvHRenderer GetHandle ()

6.21.1 Detailed Description

6.21.2 Function Documentation

6.21.2.1 LvStatus CanDisplayImage (LvBuffer^ *Buffer,* UInt32 *RenderFlags*)

Checks, if the image can be displayed. Namely the possibility to convert the image to desired display pixel format is checked.

Parameters

Buffer	Pointer to the LvBuffer to be displayed.
RenderFlags	Zero or a combination of LvRenderFlags.

Returns

Returns the LvStatus value; the value LVSTATUS_OK indicates the display is possible, the value LVSTAT↩ US_DISPLAY_CANNOT_DISPLAY indicates impossibility of pixel format conversion or a misconfiguration of the renderer. See SynView .Net Class Library Status Definitions.

6.21.2.2 LvStatus CanDisplayImage (LvBuffer^ *Buffer*)

Checks, if the image can be displayed. Namely the possibility to convert the image to desired display pixel format is checked.

Parameters

Buffer	Pointer to the LvBuffer to be displayed.

Returns

Returns the LvStatus value; the value LVSTATUS_OK indicates the display is possible, the value LVSTAT↩ US_DISPLAY_CANNOT_DISPLAY indicates impossibility of pixel format conversion or a misconfiguration of the renderer. See SynView .Net Class Library Status Definitions.

6.21.2.3 static LvStatus Close (LvRenderer^% *Renderer*) [static]

Deletes the LvRenderer class instance.

Parameters

Renderer	A pointer to the LvRenderer instance, obtained from the LvRenderer::Open() function. This pointer is set to nullptr(C++)/null(C#)/Nothing(VB) after close.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::CloseRenderer().

6.21.2.4 LvStatus DisplayImage (LvBuffer^ *Buffer,* UInt32 *RenderFlags*)

Displays the image. The image display mode is set by Renderer features, see LvRendererFtr.

Parameters

Buffer	Pointer to the LvBuffer to be displayed.
RenderFlags	Zero or a combination of LvRenderFlags.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.21.2.5 LvStatus DisplayImage (LvBuffer^ *Buffer*)

Displays the image. The image display mode is set by Renderer features, see LvRendererFtr.

Parameters

Buffer	Pointer to the LvBuffer to be displayed.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.21.2.6 LvHRenderer GetHandle ( )

Returns a handle of the Renderer (used in the Plain C API), associated with this class.

Returns

The Plain C API handle.

6.21.2.7 static LvStatus Open (LvStream^ *Stream,* LvRenderer^% *Renderer*) [static]

Creates the LvRenderer class instance for image display. The renderer attepts to load the lv.synview.display library. In case of SynView installation in an operating system without possibility to graphically display (for example Linux without XWindows), the load of this library fails and the calls to Renderer functions will return errors.

Parameters

Stream	A pointer to the LvStream instance, obtained from the LvStream::Open() function.
Renderer	In this parameter the pointer to the LvRenderer is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

See also

LvStream::OpenRenderer().

6.21.2.8 LvStatus Repaint (UInt32 *RenderFlags*)

Repaints the contents of the display window. In order to be able to repaint, all images to be displayed must be still held by the application, i.e. must not be returned to the input buffer pool. See also LvStream::LvPostponeQueue↩ Buffers feature. A typical usage of this function is in the WM_PAINT handler in a Windows application.

Parameters

RenderFlags	Zero or a combination of LvRenderFlags.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.21.2.9 LvStatus Repaint ( )

Repaints the contents of the display window. In order to be able to repaint, all images to be displayed must be still held by the application, i.e. must not be returned to the input buffer pool. See also LvStream::LvPostponeQueue↩ Buffers feature. A typical usage of this function is in the WM_PAINT handler in a Windows application.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.21.2.10 LvStatus SetWindow (IntPtr *hWnd*)

Sets the target window, in which the renderer has to display. Note that the application itself assure any repainting (when the window need to be repainted due to a movement of overlapping) - use LvRenderer::Repaint() in such case.

Parameters

hWnd	Handle to the window.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22 SynView LvModule class

Functions

- LvStatus GetNumFeatures (LvFtrGroup FtrGroup, UInt32%NumFeatures)
- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvFeature%Feature)
- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvFeature%Feature, UInt32%Level)
- LvStatus GetFeatureByName (LvFtrGroup FtrGroup, String^Name, LvFeature%Feature)
- Boolean IsImplemented (LvFeature Feature)
- Boolean IsImplementedByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsAvailable (LvFeature Feature)
- Boolean IsAvailableByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsReadable (LvFeature Feature)
- Boolean IsWritable (LvFeature Feature)
- Boolean IsAvailableEnumEntry (LvFeature Feature, LvEnum EnumEntry)
- Boolean IsImplementedEnumEntry (LvFeature Feature, LvEnum EnumEntry)
- LvStatus GetType (LvFeature Feature, LvFtrType%FtrType)
- LvStatus GetType (LvFeature Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup)
- LvStatus GetBool (LvFeature Feature, Boolean%Value)
- LvStatus SetBool (LvFeature Feature, Boolean Value)
- LvStatus GetInt32 (LvFeature Feature, Int32%Value)
- LvStatus SetInt32 (LvFeature Feature, Int32 Value)
- LvStatus GetInt32Range (LvFeature Feature, Int32%MinValue, Int32%MaxValue)
- LvStatus GetInt32Range (LvFeature Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment)
- LvStatus GetInt64 (LvFeature Feature, Int64%Value)
- LvStatus SetInt64 (LvFeature Feature, Int64 Value)
- LvStatus GetInt64Range (LvFeature Feature, Int64%MinValue, Int64%MaxValue)
- LvStatus GetInt64Range (LvFeature Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment)
- LvStatus GetInt (LvFeature Feature, Int64%Value)
- LvStatus SetInt (LvFeature Feature, Int64 Value)
- LvStatus GetIntRange (LvFeature Feature, Int64%MinValue, Int64%MaxValue)
- LvStatus GetIntRange (LvFeature Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment)
- LvStatus GetFloat (LvFeature Feature, Double%Value)
- LvStatus SetFloat (LvFeature Feature, Double Value)
- LvStatus GetFloatRange (LvFeature Feature, Double%MinValue, Double%MaxValue)
- LvStatus GetFloatRange (LvFeature Feature, Double%MinValue, Double%MaxValue, Double%Increment)
- LvStatus GetString (LvFeature Feature, String^%Value)
- LvStatus SetString (LvFeature Feature, String^Value)
- LvStatus GetBuffer (LvFeature Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetBufferSize (LvFeature Feature, SizeT%Size)
- LvStatus SetBuffer (LvFeature Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetPtr (LvFeature Feature, IntPtr%pValue)
- LvStatus SetPtr (LvFeature Feature, IntPtr pValue)
- LvStatus GetEnum (LvFeature Feature, LvEnum%Value)
- LvStatus SetEnum (LvFeature Feature, LvEnum Value)
- LvStatus GetEnumStr (LvFeature Feature, String^%SymbolicName)
- LvStatus SetEnumStr (LvFeature Feature, String^SymbolicName)
- LvStatus GetEnumValByStr (LvFeature Feature, String^SymbolicName, LvEnum%Value)
- LvStatus GetEnumValByStr (LvFeature Feature, String^SymbolicName, LvEnum%Value, LvFtrAccess%Ftr↩ Access)
- LvStatus GetEnumStrByVal (LvFeature Feature, LvEnum Value, String^%SymbolicName)
- LvStatus GetEnumStrByVal (LvFeature Feature, LvEnum Value, String^%SymbolicName, LvFtr↩ Access%FtrAccess)
- LvStatus CmdExecute (LvFeature Feature)

- LvStatus CmdExecute (LvFeature Feature, UInt32 Timeout)
- LvStatus CmdIsDone (LvFeature Feature, Boolean%IsDone)
- LvStatus GetAccess (LvFeature Feature, LvFtrAccess%FtrAccess)
- LvStatus GetVisibility (LvFeature Feature, LvFtrVisibility%FtrVisibility)
- LvStatus GetInfo (LvFeature Feature, LvFtrInfo FtrInfo, Int32%Info)
- LvStatus GetInfo (LvFeature Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param)
- LvStatus GetInfoStr (LvFeature Feature, LvFtrInfo FtrInfo, String^%InfoStr)
- LvStatus GetInfoStr (LvFeature Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param)
- LvStatus RegisterFeatureCallback (LvFeature Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩ FeatureParam)
- LvStatus StartPollingThread (UInt32 PollingTime)
- LvStatus StartPollingThread (UInt32 PollingTime, Boolean PollChildren)
- LvStatus StopPollingThread ()
- LvStatus Poll ()

6.22.1 Detailed Description

6.22.2 Function Documentation

6.22.2.1 LvStatus CmdExecute (LvFeature *Feature*) [protected]

Executes a command.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.2 LvStatus CmdExecute (LvFeature *Feature,* UInt32 *Timeout*) [protected]

Executes a command.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Timeout	If greater than 0, the LvModule::CmdIsDone() is called in a loop to wait for the command completion, until the LvModule::CmdIsDone() returns true or the Timeout (in milliseconds) expires. If set to 0, no wait is done.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.3 LvStatus CmdIsDone (LvFeature *Feature,* Boolean% *IsDone*) [protected]

Checks if the command execution has completed.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
IsDone	In this parameter is returned true, if the command is completed, otherwise false.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.4 LvStatus GetAccess (LvFeature *Feature,* LvFtrAccess% *FtrAccess*) [protected]

Gets the access mode of the feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrAccess	The access is returned in this parameter. One of the LvFtrAccess.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.5 LvStatus GetBool (LvFeature *Feature,* Boolean% *Value*) [protected]

Gets a Boolean value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The Boolean value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.6 LvStatus GetBuffer (LvFeature *Feature,* IntPtr *pBuffer,* SizeT *Size*) [protected]

Gets a block of data.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
pBuffer	Pointer to a buffer, to which the data will be stored.
Size	Size of the buffer.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.7 LvStatus GetBufferSize (LvFeature *Feature,* SizeT% *Size*) [protected]

Gets the block data size.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Size	The needed size of the buffer is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.8 LvStatus GetEnum (LvFeature *Feature,* LvEnum% *Value*) [protected]

Gets the SynView constant for the enumeration entry, if exists. If does not exist, you must work with the string enumeration entry value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The SynView constant for the enum entry is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.9 LvStatus GetEnumStr (LvFeature *Feature,* String^% *SymbolicName*) [protected]

Gets the enumeration entry as a string (symbolic name). It is not possible to get the needed size for this single feature, instead, it is possible to get the maximum size of the all enum values of this feature, by the LvModule::↩ GetInfo(LvFtrInfo::EnumEntryNameMaxSize) function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
SymbolicName	String, where the symbolic name is be returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.10 LvStatus GetEnumStrByVal (LvFeature *Feature,* LvEnum *Value,* String^% *SymbolicName*) [protected]

Returns a string symbolic name of the enum entry for the SynView constant.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Value	The SynView constant for the enum entry.
SymbolicName	String, where the symbolic name is returned. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.11 LvStatus GetEnumStrByVal (LvFeature *Feature*, LvEnum *Value*, String^% *SymbolicName*, LvFtrAccess% *FtrAccess*) [protected]

Returns a string symbolic name of the enum entry for the SynView constant.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The SynView constant for the enum entry.
SymbolicName	String, where the symbolic name is returned. Can be nullptr(C++)/null(C#)/Nothing(VB).
FtrAccess	The access mode of the enum entry is returned in this parameter - one of LvFtrAccess. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.12 LvStatus GetEnumValByStr (LvFeature *Feature*, String^ *SymbolicName*, LvEnum% *Value*) [protected]

Gets the SynView constant for the enumeration entry, if exists.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
SymbolicName	A string with symbolic name of the enum entry.
Value	The SynView constant for the enum entry is returned in this parameter. If the SynView constant does not exist for this enumeration entry, 0 is returned (no error in indicated).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.13 LvStatus GetEnumValByStr (LvFeature *Feature*, String^ *SymbolicName*, LvEnum% *Value*, LvFtrAccess% *FtrAccess*) [protected]

Gets the SynView constant for the enumeration entry, if exists.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
SymbolicName	A string with symbolic name of the enum entry.
Value	The SynView constant for the enum entry is returned in this parameter. If the SynView constant does not exist for this enumeration entry, 0 is returned (no error in indicated).
FtrAccess	The feature access is returned in this parameter - one of GroupSynView_LvFtrAccess. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.14 LvStatus GetFeatureAt (LvFtrGroup *FtrGroup*, UInt32 *Index*, LvFeature% *Feature*) [protected]

Returns the feature ID at specified position. Can be used to iterate all the features in a list.

Parameters

FtrGroup	One of the LvFtrGroup.
Index	Zero based index of the feature in the list.
Feature	Feature ID is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.15 LvStatus GetFeatureAt (LvFtrGroup *FtrGroup*, UInt32 *Index*, LvFeature% *Feature*, UInt32% *Level*) [protected]

Returns the feature ID at specified position. Can be used to iterate all the features in a list.

Parameters

FtrGroup	One of the LvFtrGroup.
Index	Zero based index of the feature in the list.
Feature	Feature ID is returned in this parameter.
Level	Feature Level expressing its position in the tree is returned in this parameter. The base level has value 1.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.16 LvStatus GetFeatureByName (LvFtrGroup *FtrGroup*, String^ *Name*, LvFeature% *Feature*) [protected]

Returns a feature ID based on the feature name. This function is a substantial function for the generic approach to the feature - by this function you can get the ID of any existing feature, that means also for those, for which a SynView constant is not defined. Be sure to check the success of this function - if the feature is not mandatory, it may not exist.

Parameters

FtrGroup	One of the LvFtrGroup.
Name	Name of the feature.
Feature	Feature ID is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.17 LvStatus GetFloat (LvFeature *Feature*, Double% *Value*) [protected]

Gets a float value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The float value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.18 LvStatus GetFloatRange (LvFeature *Feature*, Double% *MinValue*, Double% *MaxValue*) [protected]

Returns a range of a float feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.19 LvStatus GetFloatRange (LvFeature *Feature*, Double% *MinValue*, Double% *MaxValue*, Double% *Increment*) [protected]

Returns a range of a float feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
Increment	The increment value is returned in this parameter. If the increment is not defined, 0 is returned. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.20 LvStatus GetInfo (LvFeature *Feature*, LvFtrInfo *FtrInfo*, Int32% *Info*) [protected]

Gets an info in form of a 32-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrInfo	One of the LvFtrInfo.
Info	The value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.21 LvStatus GetInfo (LvFeature *Feature*, LvFtrInfo *FtrInfo*, Int32% *Info*, Int32 *Param*) [protected]

Gets an info in form of a 32-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrInfo	One of the LvFtrInfo.
Info	The value is returned in this parameter.
Param	Additional parameter, required by some types of info.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.22 LvStatus GetInfoStr (LvFeature *Feature*, LvFtrInfo *FtrInfo*, String^% *InfoStr*) [protected]

Gets an info in form of a string value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

FtrInfo	One of the LvFtrInfo.
InfoStr	The string value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.23 LvStatus GetInfoStr (LvFeature *Feature,* LvFtrInfo *FtrInfo,* String^% *InfoStr,* Int32 *Param*) [protected]

Gets an info in form of a string value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrInfo	One of the LvFtrInfo.
InfoStr	The string value is returned in this parameter.
Param	Additional parameter, required by some types of info.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.24 LvStatus GetInt (LvFeature *Feature,* Int64% *Value*) [protected]

Gets a 64-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The integer value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetInt64() function.

6.22.2.25 LvStatus GetInt32 (LvFeature *Feature,* Int32% *Value*) [protected]

Gets a 32-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The integer value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

The value is internally kept always as a 64-bit value; the functions for setting and getting a 32-bit value are provided just for convenience.

6.22.2.26 LvStatus GetInt32Range (LvFeature *Feature*, Int32% *MinValue*, Int32% *MaxValue*) [protected]

Returns a range and increment of an 32-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

The value is internally kept always as a 64-bit value; the functions for setting and getting a 32-bit value are provided just for convenience.

6.22.2.27 LvStatus GetInt32Range (LvFeature *Feature*, Int32% *MinValue*, Int32% *MaxValue*, Int32% *Increment*) [protected]

Returns a range and increment of an 32-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
Increment	The increment value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

The value is internally kept always as a 64-bit value; the functions for setting and getting a 32-bit value are provided just for convenience.

6.22.2.28 LvStatus GetInt64 (LvFeature *Feature,* Int64% *Value*) [protected]

Gets a 64-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The integer value is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetInt() function.

6.22.2.29 LvStatus GetInt64Range (LvFeature *Feature*, Int64% *MinValue*, Int64% *MaxValue*) [protected]

Returns a range and increment of an 64-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetIntRange() function.

6.22.2.30 LvStatus GetInt64Range (LvFeature *Feature*, Int64% *MinValue*, Int64% *MaxValue*, Int64% *Increment*) [protected]

Returns a range and increment of an 64-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
Increment	The increment value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetIntRange() function.

6.22.2.31 LvStatus GetIntRange (LvFeature *Feature,* Int64% *MinValue,* Int64% *MaxValue*) [protected]

Returns a range and increment of an 64-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetInt64Range() function.

6.22.2.32 LvStatus GetIntRange (LvFeature *Feature*, Int64% *MinValue*, Int64% *MaxValue*, Int64% *Increment*) [protected]

Returns a range and increment of an 64-bit integer feature.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
MinValue	The minimum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
MaxValue	The maximum value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).
Increment	The increment value is returned in this parameter. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::GetInt64Range() function.

6.22.2.33 LvStatus GetNumFeatures (LvFtrGroup *FtrGroup*, UInt32% *NumFeatures*)

Returns a number of features for specified group. This is useful for building a list of all available features (like the tree in lv.explorer).

Parameters

FtrGroup	One of the LvFtrGroup.
NumFeatures	The number of features is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.34 LvStatus GetPtr (LvFeature *Feature*, IntPtr% *pValue*) [protected]

Gets a pointer.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
pValue	The pointer is returned in this parameter.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.35 LvStatus GetString (LvFeature *Feature,* String^% *Value*) [protected]

Gets a string value. If you need first to get the string size, use the LvModule::GetStringSize() function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	String where the value is returned.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.36 LvStatus GetType (LvFeature *Feature,* LvFtrType% *FtrType*) [protected]

Returns the feature type, GUI representation and group.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrType	The feature type is returned in this parameter. The returned value is one of the LvFtrType. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.37 LvStatus GetType (LvFeature *Feature,* LvFtrType% *FtrType,* LvFtrGui% *FtrGui,* LvFtrGroup% *FtrGroup*) [protected]

Returns the feature type, GUI representation and group.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

FtrType	The feature type is returned in this parameter. The returned value is one of the LvFtrType. Can be nullptr(C++)/null(C#)/Nothing(VB).
FtrGui	The feature GUI representation is returned in this parameter. The returned value is one of the LvFtrGui. Can be nullptr(C++)/null(C#)/Nothing(VB).
FtrGroup	The feature group, to which the feature belongs. The returned value is one of the LvFtrGroup. Can be nullptr(C++)/null(C#)/Nothing(VB).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.38 LvStatus GetVisibility (LvFeature *Feature,* LvFtrVisibility% *FtrVisibility*) [protected]

Gets the feature visibility (beginner-expert-guru).

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
FtrVisibility	The visibility is returned in this parameter. One of the LvFtrVisibility.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.39 Boolean IsAvailable (LvFeature *Feature*) [protected]

A helper function, allowing simply to determine, if a feature is available. It is a wrapper around the LvModule::Get↩ Access() function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Returns

If the feature is available, returns true, otherwise false.

6.22.2.40 Boolean IsAvailableByName (LvFtrGroup *FtrGroup,* String^ *Name*) [protected]

A helper function, allowing simply to determine, if a feature is available. It is a wrapper around the LvModule::Get↩ Access() and LvModule::GetFeatureByName() functions.

Parameters

FtrGroup	One of the LvFtrGroup.
Name	Name of the feature.

Returns

If the feature is available, returns true, otherwise false.

6.22.2.41 Boolean IsAvailableEnumEntry (LvFeature *Feature,* LvEnum *EnumEntry*) [protected]

A helper function, allowing simply to determine, if an enum entry of an enum feature is available.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
EnumEntry	The SynView constant for the enum entry.

Returns

If the enum entry is available, returns true, otherwise false.

6.22.2.42 Boolean IsImplemented (LvFeature *Feature*) [protected]

A helper function, allowing simply to determine, if a feature is implemented. It is a wrapper around the LvModule↩ ::GetAccess() function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Returns

If the feature is implemented, returns true, otherwise false.

6.22.2.43 Boolean IsImplementedByName (LvFtrGroup *FtrGroup*, String^ *Name*) [protected]

A helper function, allowing simply to determine, if a feature is implemented. It is a wrapper around the LvModule↩ ::GetAccess() and LvModule::GetFeatureByName() functions.

Parameters

FtrGroup	One of the LvFtrGroup.
Name	Name of the feature.

Returns

If the feature is implemented, returns true, otherwise false.

6.22.2.44 Boolean IsImplementedEnumEntry (LvFeature *Feature*, LvEnum *EnumEntry*) [protected]

A helper function, allowing simply to determine, if an enum entry of an enum feature is implemented.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
EnumEntry	The SynView constant for the enum entry.

Returns

If the enum entry is implemented, returns true, otherwise false.

6.22.2.45 Boolean IsReadable (LvFeature *Feature*) [protected]

A helper function, allowing simply to determine, if a feature is readable. It is a wrapper around the LvModule::Get↩ Access() function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Returns

If the feature is readable, returns true, otherwise false.

6.22.2.46 Boolean IsWritable (LvFeature *Feature*) [protected]

A helper function, allowing simply to determine, if a feature is writable. It is a wrapper areound the LvModule::Get↩ Access() function.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

Returns

If the feature is writable, returns true, otherwise false.

6.22.2.47 LvStatus Poll ( )

Polls all the non-cached features of the module. If the feature polling interval expires, the value is read and the feature callback is called.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.48 LvStatus RegisterFeatureCallback (LvFeature *Feature,* Boolean *Register,* IntPtr *pUserParam,* IntPtr *pFeatureParam*) [protected]

Registers or unregisters a callback function for the feature. The callback function is created internally and transformed to the OnFeatureChanged event. This event is produced by GenApi when a feature changes its value or status. The application should process this event fast. Note that the event can be called also from another thread context - see LvEventType::FeatureDevEvent. Important note: The event handler should never set any other feature. Doing so can lead to recursions, which would be probably hard to diagnose and could cause unexpected behavior.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Register	True for registering the feature, False for unregistering it.
pUserParam	User parameter, which will be passed to the LvFeatureChangedArgs of the handler.
pFeatureParam	Second user parameter, which will be passed to the LvFeatureChangedArgs of the handler.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.49 LvStatus SetBool (LvFeature *Feature,* Boolean *Value*) [protected]

Sets a Boolean value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	Value to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.50 LvStatus SetBuffer (LvFeature *Feature*, IntPtr *pBuffer*, SizeT *Size*) [protected]

Sets a block of data.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
pBuffer	Pointer to the data.
Size	Size of the data.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.51 LvStatus SetEnum (LvFeature *Feature*, LvEnum *Value*) [protected]

Sets the enumeration entry by the SynView constant. If the SynView constant is not defined for the feature, then use LvModule::SetEnumStr() to set the enum entry by a string.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The SynView constant for the requested enumeration entry.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.52 LvStatus SetEnumStr (LvFeature *Feature*, String^ *SymbolicName*) [protected]

Sets enumeration entry by its string symbolic name.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.

SymbolicName	A string with the symbolic name of the enumeration entry.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.53 LvStatus SetFloat (LvFeature *Feature*, Double *Value*) [protected]

Sets a float value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The value to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.54 LvStatus SetInt (LvFeature *Feature*, Int64 *Value*) [protected]

Sets a 64-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	Value to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::SetInt64() function.

6.22.2.55 LvStatus SetInt32 (LvFeature *Feature*, Int32 *Value*) [protected]

Sets a 32-bit value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	Value to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

The value is internally kept always as a 64-bit value; the functions for setting and getting a 32-bit value are provided just for convenience.

6.22.2.56 LvStatus SetInt64 (LvFeature *Feature*, Int64 *Value*) [protected]

Sets a 64-bit integer value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	Value to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

This function is equal to the LvModule::SetInt() function.

6.22.2.57 LvStatus SetPtr (LvFeature *Feature*, IntPtr *pValue*) [protected]

Sets a pointer.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
pValue	The pointer to be set.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.58 LvStatus SetString (LvFeature *Feature*, String^ *Value*) [protected]

Sets a string value.

Parameters

Feature	The feature ID - use a symbolic constant (one of the SynView .Net Class Library Features) or an ID obtained by the LvModule::GetFeatureByName() function.
Value	The string value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.59 LvStatus StartPollingThread (UInt32 *PollingTime*)

Starts a thread, which in a loop polls the non-cached features. If the feature polling interval expires, the value is read and the feature callback is called.

Parameters

PollingTime	A time in milliseconds between 2 calls to poll the features.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.60 LvStatus StartPollingThread (UInt32 *PollingTime,* Boolean *PollChildren*)

Starts a thread, which in a loop polls the non-cached features. If the feature polling interval expires, the value is read and the feature callback is called.

Parameters

PollingTime	A time in milliseconds between 2 calls to poll the features.
PollChildren	If set to true, also the features in all children modules are polled. For example, if your application uses only one System module, then it is a parent of all other modules, so the polling will be propagated to all modules from a single thread. If a module has started own polling thread, then it is excluded from the propagating.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.22.2.61 LvStatus StopPollingThread ( )

Stops the polling thread. See LvModule::StartPollingThread() for details.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.23 SynView Image Processing .Net Class Library

Modules

- SynView Image Processing .Net Class Library Defines
- SynView Image Processing .Net Class Library Enums
- SynView Image Processing Library functions

6.23.1 Detailed Description

6.24 SynView Image Processing Library functions

Modules

6.24.1 Detailed Description

6.25 Common functions

6.26 Image initialization functions

Functions

- LvStatus Initialize (UInt32 Width, UInt32 Height, LvPixelFormat PixelFormat, LvipImgAttr Attributes)
- LvStatus AllocateImageData ()
- LvStatus DeallocateImageData ()
- LvStatus CopyFromBmpInfo (IntPtr pBmpInfo)
- LvStatus CopyToBmpInfo (IntPtr pBmpInfo)
- LvStatus FillWithColor (Byte Red, Byte Green, Byte Blue)
- LvStatus FillWithColor (UInt32 ColorRGB)

Variables

- property UInt32 Width
- property UInt32 Height
- property LvPixelFormat PixelFormat
- property UInt32 BytesPerPixel
- property UInt32 LinePitch
- property LvipImgAttr Attributes
- property UInt32 ImageDataSize
- property IntPtr Data
- property IntPtr ImgInfo

6.26.1 Detailed Description

6.26.2 Function Documentation

6.26.2.1 LvStatus AllocateImageData ( )

Allocates appropriate space to LvipImage::Data or color planes, according to the Height and LineIncrement.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.2 LvStatus CopyFromBmpInfo (IntPtr *pBmpInfo*)

Converts the header from Windows BITMAPINFO to LvipImage.

Parameters

pBmpInfo	pointer to BITMAPINFO

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.3 LvStatus CopyToBmpInfo (IntPtr *pBmpInfo*)

Converts LvipImage to BITMAPINFO Info. The typical usage of this function is when you need to use Windows API functions for image display; these functions usually require a pointer to BITMAPINFO and a pointer to image data. Note that Windows Device Independent Bitmap format does cover all the possible image formats of LvipImage (for example 9- to 16-bit mono formats, incompatible line increments etc.); in such case the conversion of the header fails and you will have to convert the image using the LvipConvertToPixelFormat() function first. Expects the BitmapInfo to have appropriate size, i.e. for example

```
sizeof(BITAMAPINFOHEADER) + 256*sizeof(RGBQUAD) for 8-bit image
```

Parameters

pBmpInfo	The address of the destination BITMAPINFO structure

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.4 LvStatus DeallocateImageData ( )

Deallocates the image data buffer(s) If the flags is not containing LvipImgAttr::NotDataOwner, deallocates LvipImage::Data or color planes and sets them to NULL.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.5 LvStatus FillWithColor (Byte *Red,* Byte *Green,* Byte *Blue*)

Fills image data with specified color.

Parameters

Red	8bit Red value
Green	8bit Green value
Blue	8bit Blue value

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.6 LvStatus FillWithColor (UInt32 *ColorRGB*)

Fills image data with specified color.

Parameters

ColorRGB	Color specified as Red $<<$ 16 | Green $<<$ 8 | Blue.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.2.7 LvStatus Initialize (UInt32 *Width,* UInt32 *Height,* LvPixelFormat *PixelFormat,* LvipImgAttr *Attributes*)

Initializes the LvipImgInfo to specified values, calculates the line increment and sets LvipImage::Data to NULL (be sure to deallocate the image buffers if were allocated, before this function call). If LvipImage::Data of other owner is used, set the LvipImgAttr::NotDataOwner flag so that the data are not deallocated when LvipDeallocateImageData() is applied to this ImgInfo.

Parameters

Width	Width of image in pixels
Height	Height of image in pixels
PixelFormat	Pixel format; one of the LvPixelFormat.
Attributes	Image attributes; OR-ed combination of the LvipImgAttr.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.26.3 Variable Documentation

6.26.3.1 property LvipImgAttr Attributes

Sets or returns the image attributes.

6.26.3.2 property UInt32 BytesPerPixel

Returns the pixel increment in bytes.

Returns

Returns the pixel increment.

6.26.3.3 property IntPtr Data

Sets or returns the unmanaged pointer to image data.

6.26.3.4 property UInt32 Height

Returns the height the image in pixels. To set the image size, use the LvipImage::Initialize().

Returns

Returns the height the image in pixels.

6.26.3.5 property UInt32 ImageDataSize

Returns the data size required for the image. Expects the Height and LineIncrement are already calculated. In case of color planes returns the size of one plane

Returns

The data size required for the image in bytes.

6.26.3.6 property IntPtr ImgInfo

Returns the unmanaged pointer to the ImgInfo structure.

6.26.3.7 property UInt32 LinePitch

Returns the line increment in bytes.

Returns

Returns the line increment.

6.26.3.8 property LvPixelFormat PixelFormat

Returns the pixel format of the image. To set the image pixel format, use the LvipImage::Initialize().

Returns

Returns the pixel format of the image.

6.26.3.9 property UInt32 Width

Returns the width the image in pixels. To set the image size, use the LvipImage::Initialize().

Returns

Returns the width the image in pixels.

6.27 Region of Interest (ROI) functions

Functions

- LvStatus CopyArea (LvipImage^ DstImage, Int32 DstXOffset, Int32 DstYOffset, Int32 DstWidth, Int32 DstHeight, LvipOption Options)

6.27.1 Detailed Description

6.27.2 Function Documentation

6.27.2.1 LvStatus CopyArea (LvipImage^ *DstImage,* Int32 *DstXOffset,* Int32 *DstYOffset,* Int32 *DstWidth,* Int32 *DstHeight,* LvipOption *Options*)

Extracts from the source bitmap a rectangle as destination bitmap. If the rectangle goes outside of the source image, the intersection rectangle is taken, that means the result width and/or height can be smaller than required. If the intersection is zero, the function returns returns LVSTATUS_LVIP_DST_RECT_OUTSIDE_SRC.

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstXOffset	Left offset of the rectangle
DstYOffset	Upper offset of the rectangle
DstWidth	Width of area which has to be copied
DstHeight	Height of area which has to be copied
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28 Lookup Table (LUT) functions

Functions

- LvipLut (LvipLutType LutType)
- ~LvipLut ()
- LvStatus ResetLut ()
- LvStatus Set8BitLut (array< Byte >^Red, array< Byte >^Green, array< Byte >^Blue)
- LvStatus Get8BitLut (array< Byte >^Red, array< Byte >^Green, array< Byte >^Blue)
- LvStatus Set10BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Get10BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Set12BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Get12BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Set8BitLutValue (LvEnum LutSelector, UInt32 Index, Byte Value)
- LvStatus Get8BitLutValue (LvEnum LutSelector, UInt32 Index, Byte%Value)
- LvStatus Set10BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16 Value)
- LvStatus Get10BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16%Value)
- LvStatus Set12BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16 Value)
- LvStatus Get12BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16%Value)
- LvStatus AddGammaToLut (Double Gamma)
- LvStatus AddWbToLut (Double FactorRed, Double FactorGreen, Double FactorBlue)
- LvStatus AddOffsetAndGainToLut (Double Offset, Double Gain)
- LvStatus AddBrightnessAndContrastToLut (Double Brightness, Double Contrast)
- LvStatus ApplyLut (LvipImage^DstImage, LvipOption Options, LvipLut^Lut)
- LvStatus ApplyLut (LvipLut^Lut)
- LvStatus CalcWbFactors (UInt32%FactorRed, UInt32%FactorGreen, UInt32%FactorBlue)
- LvStatus CalcWbFactors (UInt32%FactorRed, UInt32%FactorGreen, UInt32%FactorBlue, LvipOption Options)

6.28.1 Detailed Description

6.28.2 Function Documentation

6.28.2.1 LvStatus AddBrightnessAndContrastToLut (Double *Brightness,* Double *Contrast*)

Adds a brightness and contrast to the LUT. Recalculates each value in the LUT table by adding the brightness and multiplying by contrast. This function is similar to the LvipLut::AddOffsetAndGainToLut() function, with the following 2 differences:

- The Brightness middle value is 1.0, meaning no change. The Brightness 0 means black image and 2.0 means fully white image, because subtracting or adding the 1.0 means subtracting or adding the maximum pixel value.
- The Brightness factor is internally corrected in dependence on contrast. The Contrast is equivalent to Gain in the LvipLut::AddOffsetAndGainToLut() function. It is a factor, i.e. 1.0 means no change. Can be also negative -1.0 makes an inversion.

Parameters

Brightness	The Brightness to be added. See the explanation above.
Contrast	The Contrast factor.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.2 LvStatus AddGammaToLut (Double *Gamma*)

Adds gamma to LUT. Recalculates each value in the LUT table by applying the Gamma curve factor to each LUT item.

Parameters

Gamma	The base value 1.0, which means no LUT change. Values below 1 make the middle tones darker, values above 1 make the middle tones lighter. The minimal gamma value is 0.01.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.3 LvStatus AddOffsetAndGainToLut (Double *Offset,* Double *Gain*)

Adds an offset and gain to the LUT. Recalculates each value in the LUT table by adding the offset and multiplying by a gain factor. The offset is in range -1.0 to +1.0, where 0 means no change and 1.0 the maximum pixel value - adding 1.0 will make the image fully white, adding -1.0 will make it fully black. The offset is corresponding to Brightness - 1.0, see LvipLut::AddBrightnessAndContrastToLut().

The gain is a factor, by which is multiplied each LUT value (so 1.0 means no change). Can be also negative -1.0 makes an inversion. It is equivalent to contrast.

Parameters

Offset	The Offset to be added. See the explanation above.
Gain	The Gain factor.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.4 LvStatus AddWbToLut (Double *FactorRed,* Double *FactorGreen,* Double *FactorBlue*)

Adds a white balance to LUT. Recalculates each value in the LUT table by applying the white balance factors. See LvipImage::CalcWbFactors() for obtaining the WB factors from an image.

Parameters

FactorRed	Red factor of white balance. A value 1.0 means no change.
FactorGreen	Green factor of white balance. A value 1.0 means no change.
FactorBlue	Blue factor of white balance. A value 1.0 means no change.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.5 LvStatus ApplyLut (LvipImage^ *DstImage,* LvipOption *Options,* LvipLut^ *Lut*)

Apply LUT to source image and save it in the destination image. Applies the LUT to the image. Note that the LUT can be applied in other functions as well, which is faster than this separate processing.

Supported input pixel formats: 8-bit mono, 10-bit mono, 12-bit mono, 24-bit RGB, 32-bit RGB.
Supported output pixel formats: equal to the input pixel format.
Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	LUT, which has to be applied

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.6 LvStatus ApplyLut (LvipLut^ *Lut*)

Apply LUT to source image and save it in the destination image. Applies the LUT to the image. Note that the LUT can be applied in other functions as well, which is faster than this separate processing.

Supported input pixel formats: 8-bit mono, 10-bit mono, 12-bit mono, 24-bit RGB, 32-bit RGB.
Supported output pixel formats: equal to the input pixel format.
Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

Lut	LUT, which has to be applied

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.7 LvStatus CalcWbFactors (UInt32% *FactorRed,* UInt32% *FactorGreen,* UInt32% *FactorBlue*)

Calculates white balance factors. The image is expected to be obtained from camera pointed at a neutral grey area. The factor is a gain applied to each pixel component. The gain = 1.0 means no change. In order to avoid using float numbers, the factors are multiplied by 1000 and stored in uint32_t. If the image pixel format is MONO, the image is expected to be Bayer Array encoded. The factors are normalized, so that all are >= 1.0. This assures the areas with saturated colors remain white.
The obtained factors could be used in the LvipAddWbToLut() function.

Note

If the WbCorrectFactors "LvipOption::WbCorrectFactors" flag is used, it is assumed that the white balance is calculated from the image to which were applied white balancing factors passed as parameters. Thus only a correction is calculated and the existing factors are modified. This flag cannot be used on undecoded Bayer array image.

Parameters

FactorRed	Pointer to uint32_t variable to which will be saved the Red factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.

FactorGreen	Pointer to uint32_t variable to which will be saved the Green factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.
FactorBlue	Pointer to uint32_t variable to which will be saved the Blue factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.8 LvStatus CalcWbFactors (UInt32% *FactorRed,* UInt32% *FactorGreen,* UInt32% *FactorBlue,* LvipOption *Options*)

Calculates white balance factors. The image is expected to be obtained from camera pointed at a neutral grey area. The factor is a gain applied to each pixel component. The gain = 1.0 means no change. In order to avoid using float numbers, the factors are multiplied by 1000 and stored in uint32_t. If the image pixel format is MONO, the image is expected to be Bayer Array encoded. The factors are normalized, so that all are >= 1.0. This assures the areas with saturated colors remain white.
The obtained factors could be used in the LvipAddWbToLut() function.

Note

If the WbCorrectFactors "LvipOption::WbCorrectFactors" flag is used, it is assumed that the white balance is calculated from the image to which were applied white balancing factors passed as parameters. Thus only a correction is calculated and the existing factors are modified. This flag cannot be used on undecoded Bayer array image.

Parameters

FactorRed	Pointer to uint32_t variable to which will be saved the Red factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.
FactorGreen	Pointer to uint32_t variable to which will be saved the Green factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.
FactorBlue	Pointer to uint32_t variable to which will be saved the Blue factor, multiplied by 1000. If the LvipOption::WbCorrectFactors flag is used, the variable should contain the factor already used for WB of the current image.
Options	Options, see LvipOption::WbCorrectFactors

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.9 LvStatus Get10BitLut (array< UInt16 >^ *Red,* array< UInt16 >^ *Green,* array< UInt16 >^ *Blue*)

Gets 10-bit LUT data. This function fills up supplied arrays with the current LUT data. It is useful for example after calling LvipLut::AddGammaToLut() or LvipLut::AddWbToLut() to get the values of current LUT.

Parameters

Red	pointer to an array of 1024 UInt16 values, will be filled with red
Green	pointer to an array of 1024 UInt16 values, will be filled with green
Blue	pointer to an array of 1024 UInt16 values, will be filled with blue

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.10 LvStatus Get10BitLutValue (LvEnum *LutSelector,* UInt32 *Index,* UInt16% *Value*)

Gets one 10-bit LUT value. This function reads one value from the LUT. Note that for reading the whole LUT a more effective function LvipLut::Get10BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.
Value	The variable which receives the value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.11 LvStatus Get12BitLut (array< UInt16 >^ *Red,* array< UInt16 >^ *Green,* array< UInt16 >^ *Blue*)

Gets 12-bit LUT data. This function fills up supplied arrays with the current LUT data. It is useful for example after calling LvipLut::AddGammaToLut() or LvipLut::AddWbToLut() to get the values of current LUT.

Parameters

Red	pointer to an array of 4096 UInt16 values, will be filled with red
Green	pointer to an array of 4096 UInt16 values, will be filled with green
Blue	pointer to an array of 4096 UInt16 values, will be filled with blue

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.12 LvStatus Get12BitLutValue (LvEnum *LutSelector,* UInt32 *Index,* UInt16% *Value*)

Gets one 12-bit LUT value. This function reads one value from the LUT. Note that for reading the whole LUT a more effective function LvipLut::Get12BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.

Value	The variable which receives the value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.13 LvStatus Get8BitLut (array< Byte >^ *Red,* array< Byte >^ *Green,* array< Byte >^ *Blue*)

Gets 8-bit LUT data. This function fills up supplied arrays with the current LUT data. It is useful for example after calling LvipLut::AddGammaToLut() or LvipLut::AddWbToLut() to get the values of current LUT.

Parameters

Red	pointer to an array of 256 Byte values, which will be filled with the Red LUT values
Green	pointer to an array of 256 Byte values, which will be filled with the Green LUT values
Blue	pointer to an array of 256 Byte values, which will be filled with the Blue LUT values

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.14 LvStatus Get8BitLutValue (LvEnum *LutSelector,* UInt32 *Index,* Byte% *Value*)

Gets one 8-bit LUT value. This function reads one value from the LUT. Note that for reading the whole LUT a more effective function LvipLut::Get8BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.
Value	The variable which receives the value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.15 LvipLut (LvipLutType *LutType*)

Constructs the class and allocates the internal LUT structures according to the specified LvipLutType.

Parameters

LutType	type of LUT which has to be allocated. One of LvipLutType.

6.28.2.16 LvStatus ResetLut ( )

Resets the LUT data to the linear order.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.17 LvStatus Set10BitLut (array< UInt16 >^ *Red,* array< UInt16 >^ *Green,* array< UInt16 >^ *Blue*)

Sets up 10-bit LUT data. Sets the LUT from 3 arrays of 1024 UInt16 values with 10-bit values. For processing the monochrome images only the green is used.

Parameters

Red	pointer to an array of 1024 UInt16 red LUT values
Green	pointer to an array of 1024 UInt16 green LUT values
Blue	pointer to an array of 1024 UInt16 blue LUT values

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.18 LvStatus Set10BitLutValue (LvEnum *LutSelector,* UInt32 *Index,* UInt16 *Value*)

Sets one 10-bit LUT value. This function writes one value to the LUT. Note that for writing the whole LUT a more effective function LvipLut::Set10BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.
Value	The value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.19 LvStatus Set12BitLut (array< UInt16 >^ *Red,* array< UInt16 >^ *Green,* array< UInt16 >^ *Blue*)

Sets up 12-bit LUT data. Sets the LUT from 3 arrays of 4096 UInt16 values with 12-bit values. For processing the monochrome images only the green is used.

Parameters

Red	pointer to an array of 4096 UInt16 red LUT values
Green	pointer to an array of 4096 UInt16 green LUT values
Blue	pointer to an array of 4096 UInt16 blue LUT values

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.20 LvStatus Set12BitLutValue (LvEnum *LutSelector,* UInt32 *Index,* UInt16 *Value*)

Sets one 12-bit LUT value. This function writes one value to the LUT. Note that for writing the whole LUT a more effective function LvipLut::Set12BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.

Value	The value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.21 LvStatus Set8BitLut (array< Byte >^ *Red*, array< Byte >^ *Green*, array< Byte >^ *Blue*)

Sets up the 8-bit LUT data. Sets the LUT from 3 arrays of 256 Byte values. For processing the monochrome images only the green is used.

Parameters

Red	pointer to an array of 256 Byte values with red LUT values
Green	pointer to an array of 256 Byte values with green LUT values
Blue	pointer to an array of 256 Byte values with blue LUT values

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.22 LvStatus Set8BitLutValue (LvEnum *LutSelector*, UInt32 *Index*, Byte *Value*)

Sets one 8-bit LUT value. This function writes one value to the LUT. Note that for writing the whole LUT a more effective function LvipLut::Set8BitLut() is available.

Parameters

LutSelector	LUT selector (see LvLUTSelector). The Luminance LUT is actually stored in the Green one.
Index	Value index in the LUT.
Value	The value.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.28.2.23 ~LvipLut ( )

Destructs the class and deallocates the internal LUT structures.

6.29 Bayer decoding/encoding functions

Functions

- LvStatus BdShowMosaic (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdGreenToGreyscale (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdNearestNeighbour (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdNearestNeighbour (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdBilinearInterpolation (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdBilinearColorCorrection (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdVariableGradients (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdVariableGradients (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdPixelGrouping (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdEncodeToBayer (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)

6.29.1 Detailed Description

6.29.2 Function Documentation

6.29.2.1 LvStatus BdBilinearColorCorrection (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

Bayer Decoding: The Bilinear interpolation with Linear Color Correction method The interpolation with Linear Color Correction (LCC) is another adaptive algorithm and optimized for images with edges in horizontal and vertical direction.

Note

This function does not support LUT due to the 2-pass algorithm

Parameters

DstImage	Destination image
DstPixelFormat	to which LvPixelFormat has to be image converted
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.2 LvStatus BdBilinearInterpolation (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options,* LvipLut^ *Lut*)

Bayer Decoding: The Bilinear Interpolation method The most commonly used method for fast Bayer decoding. For the color not directly available for the given pixel makes the linear interpolation between the 2 or 4 neigbouring pixels to get it. Gives good results with a high speed.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 8-bit mono, 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.3 LvStatus BdEncodeToBayer (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

This function encode an RGB image back to a Bayer encoded image. This function is generally for testing purposes.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat convert
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.4 LvStatus BdGreenToGreyscale (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

Bayer Decoding: Convert green to greyscale Converts fast but roughly the Bayer encoded image to a greyscale by using only the green pixels, which cover the half of all pixels. The other half is calculated by bilinear interpolation.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 8-bit mono.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	destination LvPixelFormat
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.5 LvStatus BdNearestNeighbour (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options,* LvipLut^ *Lut*)

Bayer Decoding: The Nearest Neigbour method The fastest method for Bayer array decoding. It uses the nearest pixel with the required lense color to get the pixel value. Gives rough results.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 8-bit mono, 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.6 LvStatus BdNearestNeighbour (LvipImage^ *DstImage*, LvPixelFormat *DstPixelFormat*, LvipOption *Options*)

Bayer Decoding: The Nearest Neigbour method The fastest method for Bayer array decoding. It uses the nearest pixel with the required lense color to get the pixel value. Gives rough results.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 8-bit mono, 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.7 LvStatus BdPixelGrouping (LvipImage^ *DstImage*, LvPixelFormat *DstPixelFormat*, LvipOption *Options*)

Bayer Decoding: The Pixel Grouping method. A method similar to the LvipBdVariableGradients(), but simplified and thus faster, still giving very good results.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Note

This function does not support LUT operations because of too high CPU load

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat convert
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.8 LvStatus BdShowMosaic (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

Bayer Decoding: Show mosaic. This function converts the Bayer encoded image to RGB format, without decoding the color information, only showing in the color how the image is seen by the chip after the light goes through the color lenses. The purpose of this function is only to help in Bayer decoding algorithms investigations.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat convert bayer encoded image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.9 LvStatus BdVariableGradients (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

Bayer Decoding: Variable gradients method One of the best known methods for Bayer decoding, but significantly slower than the bilinear interpolation. It is based on evaluation the color gradients in 8 directions around the pixel and selecting the set of best set for the color interpolation.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat convert

Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.29.2.10 LvStatus BdVariableGradients (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options,* LvipLut^ *Lut*)

Bayer Decoding: Variable gradients method One of the best known methods for Bayer decoding, but significantly slower than the bilinear interpolation. It is based on evaluation the color gradients in 8 directions around the pixel and selecting the set of best set for the color interpolation.

- Supported input pixel formats: 8-bit mono.
- Supported output pixel formats: 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	To which LvPixelFormat convert
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.30 Rotation and line manipulation functions

Functions

- LvStatus Deinterlace (LvipImage^ DstImage, LvipOption Options)
- LvStatus Rotate90 (LvipImage^ DstImage, Boolean ClockWise, LvipOption Options, LvipLut^ Lut)
- LvStatus Mirror (LvipImage^ DstImage, Boolean TopBottomMirror, Boolean LeftRightMirror, LvipOption Options, LvipLut^ Lut)
- LvStatus Rotate90AndMirror (LvipImage^ DstImage, Boolean ClockWise, Boolean TopBottomMirror, Boolean LeftRightMirror, LvipOption Options, LvipLut^ Lut)
- LvStatus ReverseLines ()
- LvStatus ReverseLines (LvipImage^ DstImage, LvipOption Options)

6.30.1 Detailed Description

6.30.2 Function Documentation

6.30.2.1 LvStatus Deinterlace (LvipImage^ *DstImage,* LvipOption *Options*)

Deinterlacing. Deinterlaces by averaging the neighbour lines. Deinterlace function reduces the artefacts resulting from capturing a moving object by an interlaced camera.

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.30.2.2 LvStatus Mirror (LvipImage^ *DstImage,* Boolean *TopBottomMirror,* Boolean *LeftRightMirror,* LvipOption *Options,* LvipLut^ *Lut*)

Mirrors the image along the horizontal axis (TopBottomMirror) or vertical axis (LeftRightMirror).

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image.
TopBottomMirror	**1** for top-bottom mirror, **0** if not
LeftRightMirror	**1** for left-right mirror, **0** if not
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.30.2.3 LvStatus ReverseLines ( )

Reversed lines for switching between the top-down and bottom-up formats. Performs the same action as Top↩ Bottom mirror, but updates also the LvipImage with a flag indicating the orientation (this has a meaning when switching between top-down and bottom-up formats).

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.30.2.4 LvStatus ReverseLines (LvipImage^ *DstImage,* LvipOption *Options*)

Reversed lines for switching between the top-down and bottom-up formats. Performs the same action as Top↩ Bottom mirror, but updates also the LvipImage with a flag indicating the orientation (this has a meaning when switching between top-down and bottom-up formats).

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.30.2.5 LvStatus Rotate90 (LvipImage^ *DstImage,* Boolean *ClockWise,* LvipOption *Options,* LvipLut^ *Lut*)

Rotates the image by 90 degrees clockwise or counterclockwise.

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: No.

Note

the LUT in this function is not yet implemented.
For 180 degrees rotation use the LvipMirror() function and set mirroring along both axes.

Parameters

DstImage	Destination image
ClockWise	**1** if the image has to be rotated clockwise, **0** if the image has to be rotated by counterclockwise
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.30.2.6 LvStatus Rotate90AndMirror (LvipImage^ *DstImage,* Boolean *ClockWise,* Boolean *TopBottomMirror,* Boolean *LeftRightMirror,* LvipOption *Options,* LvipLut^ *Lut*)

It does the rotation and mirroring in the same step. If the Options contain LvipOption::ReallocateDst and the Dst↩ Image contains different image width or height or the LvipImage::Data is NULL, the LvipImage::Data is reallocated and the image parameters are adjusted. The Options in such case can contain also LvipImgAttr flags for new image descriptor creation.

- Supported input pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: No.

Parameters

DstImage	Destination image
ClockWise	**1** if image has to be rotated by 90 degrees clockwise, otherwise (counterclockwise) **0**
TopBottomMirror	**1** if top-bottom mirror has to be used, otherwise **0**
LeftRightMirror	**1** if left-right mirror has to be used, otherwise **0**
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.31 Pixel format conversion functions

Functions

- LvStatus ConvertToPixelFormat (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)

6.31.1 Detailed Description

6.31.2 Function Documentation

6.31.2.1 LvStatus ConvertToPixelFormat (LvipImage^ *DstImage,* LvPixelFormat *DstPixelFormat,* LvipOption *Options*)

Converts the image from one pixel format to another one.

- Supported input pixel formats: 8-bit to 16-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: 8-bit mono, 24-bit RGB, 32-bit RGB.
- Can be done in-place: No.

Parameters

DstImage	Destination image
DstPixelFormat	Destination LvPixelFormat
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.32 Saving/loading functions

Functions

- LvStatus SaveToTiff (String^ FileName, LvipOption Options)
- LvStatus LoadFromTiff (String^ FileName, LvipOption Options)
- LvStatus SaveToBmp (String^ FileName, LvipOption Options)
- LvStatus LoadFromBmp (String^ FileName, LvipOption Options)
- LvStatus SaveToJpeg (String^ FileName, UInt32 QualityFactor)
- LvStatus LoadFromJpeg (String^ FileName, LvipOption Options)

6.32.1 Detailed Description

6.32.2 Function Documentation

6.32.2.1 LvStatus LoadFromBmp (String^ *FileName,* LvipOption *Options*)

Loads image from BMP file. Formats with less 8 bits per pixel are not supported. The color palette by 8-bit pixel format is expected to be greyscale.

Parameters

FileName	File name
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.32.2.2 LvStatus LoadFromJpeg (String^ *FileName,* LvipOption *Options*)

Loads the image from JPEG file.

Parameters

FileName	File name.
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image. See LvipOption::JpegConvertToBgr, LvipOption::JpegReadHeaderOnly and LvipOption::ReallocateDst.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

You can either supply the pImgInfo with already allocated buffer or use empty ImgInfo and specify the LvipOption::ReallocateDst flag. In the first case you can utilize the LvipOption::JpegReadHeaderOnly flag to obtain the image attributes.

6.32.2.3 LvStatus LoadFromTiff (String^ *FileName,* LvipOption *Options*)

Loads the image from TIFF file. Is preferred to load the image from TIFF file which had been previously saved by LvipSaveToTiff() function - this library supports only a base TIFF format and there it is not assured that the TIFF image created by another application could be loaded without error.

Parameters

FileName	File name
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

The LvipImage::Data are always reallocated.

6.32.2.4 LvStatus SaveToBmp (String^ *FileName,* LvipOption *Options*)

Saves the image to a BMP file if the pixel format is compatible with BMP. *Compatible with BMP* means that LvPixelFormat is one of 8-bit mono, 24- or 32-bit RGB.

- Supported pixel formats: 8-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.

Parameters

FileName	File name
Options	Options - OR-ed combination of LvipOption.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.32.2.5 LvStatus SaveToJpeg (String^ *FileName,* UInt32 *QualityFactor*)

Saves the image to the JPEG file. In contrast to the BMP format, it enables to store also 9- to 16-bit mono formats.

- Supported pixel formats: 8-bit to 16-bit mono, all RGB and BGR formats. For JPEG the native pixel format is either 8-bit mono or 24-bit RGB. If the image is in different pixel format, it is automatically converted to one of these 2 formats.

Parameters

FileName	File name.

QualityFactor	The quality factor in range from 0 to 100. The higher the quality, the lower is the compression. The default quality is 75.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.32.2.6 LvStatus SaveToTiff (String^ *FileName,* LvipOption *Options*)

Saves the image to the TIFF file. In contrast to the BMP format, it enables to store also 9- to 16-bit mono formats. The flag LvipOption::TiffConvertTo16Bit can be used to force the conversion to 16bit format, which is supported by wider range of applications.

- Supported pixel formats: 8-bit to 16-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.

Parameters

FileName	File name
Options	Options - OR-ed combination of LvipOption.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.33 Overlay functions

Functions

- LvipOverlay (UInt32 Width, UInt32 Height, LvPixelFormat PixelFormat)
- ~LvipOverlay ()
- LvStatus Paint (LvipImage^ DstImage, Int32 XPos, Int32 YPos)
- LvStatus Wipe (UInt32 ColorRGB)
- LvStatus SetTransparentColor (UInt32 ColorRGB)
- UInt32 GetTransparentColor ()
- IntPtr GetDc ()
- LvStatus ReleaseDc ()
- LvStatus SetTextParams (String^ Font, Int32 Size, UInt32 Color, UInt32 OutlineColor, LvipTextAttr Attributes, UInt32 CharSet)
- LvStatus WriteText (String^ Text, Int32 XOffset, Int32 YOffset)
- LvStatus PutBitmap (LvipImage^ Image, Int32 XOffset, Int32 YOffset)
- LvStatus PutBitmapFromBmpFile (String^ FileName, Int32 XOffset, Int32 YOffset)

Variables

- property UInt32 Width
- property UInt32 Height
- property LvPixelFormat PixelFormat

6.33.1 Detailed Description

6.33.2 Function Documentation

6.33.2.1 IntPtr GetDc ( )

Returns Windows device context for painting on the overlay. Returns Windows device context for painting on the overlay. Thus you can use Windows GDI functions, like MoveToEx(), LineTo(), DrawText() etc. for free painting on the overlay object. It may be useful to wipe the overlay object before painting by the LvipWipeOverlay() function.

Be sure you do not forget to free the device context by the LvipReleaseOverlayDc() function after the painting is done.

Returns

The device context of the overlay object, or NULL in case of failure.

Note

Available in the Windows version of the library only.

6.33.2.2 UInt32 GetTransparentColor ( )

Returns the overlay transparent color. Returns the overlay transparent color, set by the LvipSetOverlayTransparent↩ Color() function.

Returns

The transparent color as the COLORREF value or the LvipColor::None constant.

Note

Available in the Windows version of the library only.

6.33.2.3 LvipOverlay (UInt32 *Width,* UInt32 *Height,* LvPixelFormat *PixelFormat*)

Creates an overlay object. Creates an overlay object, which can be used by the following functions:

- the overlay can be filled with a color using the LvipWipeOverlay() function,
- text can be written to the overlay using the LvipWriteTextToOverlay() function,
- a bitmap can placed to the overlay using the LvipPutBitmapToOverlay() and LvipPutBitmapToOverlayFrom↩ BmpFile() functions
- Windows GDI functions can be used to paint to the overlay using the LvipGetOverlayDc() and LvipRelease↩ OverlayDc() functions,
- transparent color can be specified using the LvipSetOverlayTransparentColor() function. The overlay can be placed on an image using the LvipPaintOverlay() function. The overlay functionality is based on Windows GDI, thus the overlay functions are available only in the Windows version of the library and only Windows DIB compatible pixel formats are supported.

Parameters

Width	Width of the overlay. The smaller is the overlay, the faster is the process of combining the overlay with the destination image, so use only necessary size. The position of the overlay is specified at the time of its painting using the LvipPaintOverlay() function.
Height	Height of the overlay.
PixelFormat	The pixel format of the overlay. Must be equal to the pixel format of the destination image with which the overlay is to be combined. Only Windows DIB compatible pixel formats are supported.

Note

Available in the Windows version of the library only.

6.33.2.4 LvStatus Paint (LvpImage^ *DstImage,* Int32 *XPos,* Int32 *YPos*)

Paints the overlay at specified position. Paints the overlay at specified position in the destination image.

Parameters

DstImage	Destination image. Note that the overlay must have the same pixel format as the destination image.
XPos	X-axis position of the upper left corner of the overlay in the image.
YPos	Y-axis position of the upper left corner of the overlay in the image. Note that the upper left corner of the image has coordinates 0,0 and the X-axis goes from left to right and Y-axis from top to bottom.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions. The overlay can be placed partially or fully outside of the destination image; in such case it is clipped.

Note

Available in the Windows version of the library only.

6.33.2.5 LvStatus PutBitmap (LvpImage^ *Image,* Int32 *XOffset,* Int32 *YOffset*)

Places a bitmap to the overlay. Places a bitmap to the overlay. The bitmap must be in Windows Device Independent Bitmap (DIB) format.

Parameters

Image	Image to be placed.
XOffset	Horizontal distance of the bitmap upper left corner from the overlay rectangle left side.
YOffset	Vertical distance of the bitmap upper left corner from the overlay rectangle top side.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.6 LvStatus PutBitmapFromBmpFile (String^ *FileName,* Int32 *XOffset,* Int32 *YOffset*)

Places a bitmap from BMP file to the overlay. Places a bitmap from BMP file to the overlay. The bitmap must be in Windows Device Independent Bitmap (DIB) format.

Parameters

FileName	Name of the BMP file, including the full path.
XOffset	Horizontal distance of the bitmap upper left corner from the overlay rectangle left side.
YOffset	Vertical distance of the bitmap upper left corner from the overlay rectangle top side.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.7 LvStatus ReleaseDc ( )

Releases the Windows device context of the overlay. Releases the Windows device context of the overlay obtained by LvipGetOverlayDc() function.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.8 LvStatus SetTextParams (String^ *Font,* Int32 *Size,* UInt32 *Color,* UInt32 *OutlineColor,* LvipTextAttr *Attributes,* UInt32 *CharSet*)

Sets the parameters for the overlay text. Sets the parameters for the overlay text, which can be written to the overlay by the LvipWriteTextToOverlay() function.

Parameters

Font	Name of the font, for example "Arial".
Size	Size of the font.
Color	Color of the font, as the COLORREF value (see lvx_WipeOverlay() for details about the color values).
OutlineColor	Color for the font shadow or outline, as the COLORREF value.
Attributes	Font attributes. Use the LvipTextAttr constants to specify attributes. You can use PLUS or bitwise OR operator to combine multiple attributes.
CharSet	The character set to be used. Use Windows constants: DEFAULT_CHARSET=1, ANSI_CHARSET=0, etc.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.9 LvStatus SetTransparentColor (UInt32 *ColorRGB*)

Specifies the overlay transparent color. Specifies the transparent color. When combining the overlay with the destination image, the pixels in the overlay with the transparent color are not copied to the the destination image.

Initially, the transparent color is set to LvipColor::None, that means no overlay transparency is used.

Parameters

ColorRGB	Color defined as the COLORREF value, which has the 0x00bbggrr hexadecimal form. The low-order byte contains a value for the relative intensity of red; the second byte contains a value for green; and the third byte contains a value for blue. The high-order byte must be zero. Use LvipColor::None for disabling the transparency.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.10 LvStatus Wipe (UInt32 *ColorRGB*)

Fills the whole overlay with a specified color. Fills the whole overlay with a specified color. Initially the overalay is filled with a black color.

Parameters

ColorRGB	Color defined as the COLORREF value, which has the 0x00bbggrr hexadecimal form. The low-order byte contains a value for the relative intensity of red; the second byte contains a value for green; and the third byte contains a value for blue. The high-order byte must be zero.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.

6.33.2.11 LvStatus WriteText (String^ *Text*, Int32 *XOffset*, Int32 *YOffset*)

Writes the text to the overlay at the specified position. Writes the text to the overlay at the specified position, using the text attributes set by the LvipSetOverlayTextParams() function.

Parameters

Text	Text to be written.
XOffset	Horizontal distance of the text upper left corner from the overlay rectangle left side.
YOffset	Vertical distance of the text upper left corner from the overlay rectangle top side.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

Available in the Windows version of the library only.
If you want to change the existing text in the overlay, you must first wipe the area of the existing text and then to place the new text.

6.33.2.12 ~LvipOverlay ( )

Deletes existing object. Deletes existing overlay object identified by its handle.

Note

Available in the Windows version of the library only.

6.33.3 Variable Documentation

6.33.3.1 property UInt32 Height

Returns overlay height. Returns the overlay height specified in the LvipCreateOverlay() function, when the overlay was created.

Returns

Height of the overlay.

Note

Available in the Windows version of the library only.

6.33.3.2 property LvPixelFormat PixelFormat

Returns overlay pixel format. Returns the overlay pixel format specified in the LvipCreateOverlay() function, when the overlay was created.

Returns

Pixel format of the overlay.

Note

Available in the Windows version of the library only.

6.33.3.3 property UInt32 Width

Returns overlay width. Returns the overlay width specified in the LvipCreateOverlay() function, when the overlay was created.

Returns

Width of the overlay.

Note

Available in the Windows version of the library only.

6.34 RGB color correction and convolution functions

Functions

- LvStatus SetSaturation (Double Saturation)
- LvStatus Set3x3Sharpening (Double Factor, Boolean Full)
- LvStatus ApplyRgbColorCorrection (LvipImage^ DstImage, LvipColorCorrectionMatrix^ Matrix, LvipOption Options, LvipLut^ Lut)
- LvStatus ApplyRgbColorCorrection (LvipImage^ DstImage, LvipColorCorrectionMatrix^ Matrix, LvipOption Options)
- LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^ Matrix, LvipLut^ Lut)
- LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^ Matrix)
- LvStatus Apply3x3Convolution (LvipImage^ DstImage, LvipConvolutionMatrix^ Matrix, LvipOption Options, LvipLut^ Lut)
- LvStatus Apply3x3Convolution (LvipImage^ DstImage, LvipConvolutionMatrix^ Matrix, LvipOption Options)

6.34.1 Detailed Description

6.34.2 Function Documentation

6.34.2.1 LvStatus Apply3x3Convolution (LvipImage^ *DstImage,* LvipConvolutionMatrix^ *Matrix,* LvipOption *Options,* LvipLut^ *Lut*)

Does 3x3 convolution. Applies the 3x3 matrix convolution operation. Typically, if the matrix is set for sharpening, sharpens the image.

See also

LvipSet3x3MatrixSharpening() for creation of the sharpening matrix.

- Supported input pixel formats: 8-bit to 16-bit mono, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: No.

Parameters

DstImage	Destination image
Matrix	Matrix for the convolution operation.
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.34.2.2 LvStatus Apply3x3Convolution (LvipImage^ *DstImage,* LvipConvolutionMatrix^ *Matrix,* LvipOption *Options*)

Does 3x3 convolution. Applies the 3x3 matrix convolution operation. Typically, if the matrix is set for sharpening, sharpens the image.

See also

LvipSet3x3MatrixSharpening() for creation of the sharpening matrix.

- Supported input pixel formats: 8-bit to 16-bit mono, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: No.

Parameters

DstImage	Destination image
Matrix	Matrix for the convolution operation.
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.34.2.3 LvStatus ApplyRgbColorCorrection (LvipImage^ *DstImage,* LvipColorCorrectionMatrix^ *Matrix,* LvipOption *Options,* LvipLut^ *Lut*)

RGB color correction. A color correction 3x3 matrix is applied to RGB components of each pixel.

- Supported input pixel formats: 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image.
Matrix	Matrix used to correct colors. It could be filled up using LvipSetSaturationMatrix(). The factors in the matrix are expressed as multiplied by 1000, that means 1000 = factor 1.0.
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.34.2.4 LvStatus ApplyRgbColorCorrection (LvipImage^ *DstImage,* LvipColorCorrectionMatrix^ *Matrix,* LvipOption *Options*)

RGB color correction. A color correction 3x3 matrix is applied to RGB components of each pixel.

- Supported input pixel formats: 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image.
Matrix	Matrix used to correct colors. It could be filled up using LvipSetSaturationMatrix(). The factors in the matrix are expressed as multiplied by 1000, that means 1000 = factor 1.0.
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.34.2.5 LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^ *Matrix,* LvipLut^ *Lut*)

RGB color correction. A color correction 3x3 matrix is applied to RGB components of each pixel.

- Supported input pixel formats: 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

Matrix	Matrix used to correct colors. It could be filled up using LvipSetSaturationMatrix(). The factors in the matrix are expressed as multiplied by 1000, that means 1000 = factor 1.0.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Note

the LUT in this function is not yet implemented.

6.34.2.6 LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^ *Matrix*)

RGB color correction. A color correction 3x3 matrix is applied to RGB components of each pixel.

- Supported input pixel formats: 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

Matrix	Matrix used to correct colors. It could be filled up using LvipSetSaturationMatrix(). The factors in the matrix are expressed as multiplied by 1000, that means 1000 = factor 1.0.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.34.2.7 LvStatus Set3x3Sharpening (Double *Factor,* Boolean *Full*)

Sets up sharpening matrix. Fills the matrix with weighted values for 3x3 sharpening.
The factor is 0 for no-change matrix, +100 for maximum sharpening, -100 for blurring

Parameters

Factor	Factor of sharpening
Full	False for faster sharpening from 4 neighboring pixels, True for full sharpening from 8 neighboring pixels.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.34.2.8 LvStatus SetSaturation (Double *Saturation*)

Sets up the color saturation 3x3 matrix. The saturation factor is in percents, eg. 100 = 1.0 = unchanged image. The matrix can be used as parameter in the LvipApplyRgbColorCorrection() function.

Parameters

Saturation	the saturation factor in percents

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.35 Shading correction functions

Functions

- LvStatus ApplyShadingCorrection (LvipImage^ DstImage, LvipImage^ BlackRefImage, LvipImage^ WhiteRefImage, LvipOption Options, LvipLut^ Lut)
- LvStatus ApplyShadingCorrection (LvipImage^ DstImage, LvipImage^ BlackRefImage, LvipImage^ WhiteRefImage, LvipOption Options)

6.35.1 Detailed Description

6.35.2 Function Documentation

6.35.2.1 LvStatus ApplyShadingCorrection (LvipImage^ *DstImage,* LvipImage^ *BlackRefImage,* LvipImage^ *WhiteRefImage,* LvipOption *Options,* LvipLut^ *Lut*)

Applies the shading correction. The pBlackRefImgInfo and pWhiteRefImgInfo must be either NULL or must point to a valid image of the same pixel format as the source LvipImage.

- Supported input pixel formats: 8-bit to 16-bit mono, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image
BlackRefImage	Black reference image
WhiteRefImage	White reference image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
Lut	Lookup table

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

6.35.2.2 LvStatus ApplyShadingCorrection (LvipImage^ *DstImage,* LvipImage^ *BlackRefImage,* LvipImage^ *WhiteRefImage,* LvipOption *Options*)

Applies the shading correction. The pBlackRefImgInfo and pWhiteRefImgInfo must be either NULL or must point to a valid image of the same pixel format as the source LvipImage.

- Supported input pixel formats: 8-bit to 16-bit mono, 24-bit RGB, 32-bit RGB.
- Supported output pixel formats: equal to the input pixel format.
- Can be done in-place: Yes (DstImgInfo can be NULL).

Parameters

DstImage	Destination image
BlackRefImage	Black reference image
WhiteRefImage	White reference image
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the LvStatus value indicating the success of the requested operation. See SynView .Net Class Library Status Definitions.

Chapter 7

Namespace Documentation

7.1 NewElectronicTechnology Namespace Reference

Namespaces

- SynView

7.1.1 Detailed Description

The NewElectronicTechnology namespace This namespace is the root namespace for all New Electronic Technology .Net Class libraries.

7.2 NewElectronicTechnology::SynView Namespace Reference

Classes

- class abstract
- class LvBuffer
- class LvDevice
- class LvEvent
- class LvEventArgs
- class LvException
- class LvFeatureChangedArgs
- class LvInterface
- class LvipColorCorrectionMatrix
- class LvipConvolutionMatrix
- class LvipImage
- struct LvipImgInfo
- class LvipLut
- class LvipOverlay
- class LvNewBufferEventArgs
- class LvRenderer
- class LvStream
- class LvSystem

Typedefs

- typedef UInt32 SizeT
- typedef UInt32 LvFeature
- typedef Int32 LvStatus
- typedef UInt32 LvEnum
- typedef UInt32 LvHModule
- typedef LvHModule LvHSystem
- typedef LvHModule LvHInterface
- typedef LvHModule LvHDevice
- typedef LvHModule LvHStream
- typedef LvHModule LvHEvent
- typedef LvHModule LvHRenderer
- typedef LvHModule LvHBuffer
- typedef IntPtr **LvipHLut**
- typedef IntPtr **LvipHOverlay**

Enumerations

- enum LvStreamStartFlags : UInt32 { Default }
- enum LvStreamStopFlags : UInt32 { Default, Kill }
- enum LvUniLutFlags : UInt32 { None, HwLut }
- enum LvSaveFlag : UInt32 {
  None, RemoteFtr, LocalFtr, GenTlFtr,
  All, IgnoreVersion, IgnoreModel }
- enum LvLibInfo : LvEnum {
  BinPath, AppDataPath, UserDataPath, CfgPath,
  InstPath, IniFile, BuildDate }
- enum LvFtrGroup : LvEnum {
  DeviceRemote, SystemGtl, InterfaceGtl, DeviceGtl,
  StreamGtl, BufferGtl, SystemLocal, InterfaceLocal,
  DeviceLocal, StreamLocal, BufferLocal, RendererLocal,
  EventLocal, Buffer, Event, EventItemsGtl,
  BufferItemsGtl, SystemHidden, InterfaceHidden, DeviceHidden,
  StreamHidden, BufferHidden, RendererHidden, EventHidden }
- enum LvFtrType : LvEnum {
  Integer, Float, String, Enumeration,
  Boolean, Command, Category, StringList,
  Pointer, Buffer, Other }
- enum LvFtrGui : LvEnum {
  IntEdit, IntEditHex, IntSlider, IntSliderLog,
  FloatEdit, FloatSlider, FloatSliderLog, Label,
  StringEdit, CheckBox, ComboBox, Button,
  IpV4Address, IpMacAddress, Undefined }
- enum LvFtrVisibility : LvEnum { Beginner, Expert, Guru, Invisible }
- enum LvFtrAccess : LvEnum {
  NotImplemented, NotAvailable, WriteOnly, ReadOnly,
  ReadWrite }
- enum LvFtrInfo : LvEnum {
  IsStreamable, IsWrapped, IsSelector, IsCached,
  PollingTime, Name, DisplayName, Description,
  PhysicalUnits, ToolTip, SymbolicConst, SymbolicEnumConst,
  SelectedFeatures, SelectingFeatures, SymbolicGroupConst, ModuleName,
  FitsTo32Bit, TakeAsReadOnly, EnumEntryName, EnumEntryDisplayName,
  EnumEntryDescription, EnumEntryToolTip, EnumEntryAccess, EnumEntryValue,
  EnumEntryCount, EnumEntryNameMaxSize, InterfaceID, InterfaceDisplayName,
  InterfaceTlType, DeviceID, DeviceVendor, DeviceModel,
  DeviceTlType, DeviceDisplayName, DeviceAccessStatus }

- enum LvSystemFtr : LvFeature {
  TLVendorName, TLModelName, TLID, TLVersion,
  TLPath, TLType, GenTLVersionMajor, GenTLVersionMinor,
  GevVersionMajor, GevVersionMinor, InterfaceUpdateList, InterfaceSelector,
  InterfaceID, GevInterfaceMACAddress, GevInterfaceDefaultIPAddress, GevInterfaceDefaultSubnetMask,
  GevInterfaceDefaultGateway, LvSystemDisplayName, Info }
- enum LvInterfaceFtr : LvFeature {
  InterfaceID, InterfaceType, GevInterfaceGatewaySelector, GevInterfaceGateway,
  GevMACAddress, GevInterfaceSubnetSelector, GevInterfaceSubnetIPAddress, GevInterfaceSubnetMask,
  DeviceUpdateList, DeviceSelector, DeviceID, DeviceVendorName,
  DeviceModelName, DeviceAccessStatus, GevDeviceIPAddress, GevDeviceSubnetMask,
  GevDeviceMACAddress, LvInterface_LvDeviceUserID, LvInterface_LvDeviceSerialNumber, LvInterface↩
  DisplayName,
  Info }
- enum LvDeviceFtr : LvFeature {
  DeviceVendorName, DeviceModelName, DeviceManufacturerInfo, DeviceVersion,
  DeviceFirmwareVersion, LvRecoveryFirmwareVersion, DeviceSerialNumber, DeviceUserID,
  LvSensorID, LvGrabberID, DeviceScanType, DeviceRegistersStreamingStart,
  DeviceRegistersStreamingEnd, DeviceRegistersCheck, DeviceRegistersValid, DeviceReset,
  DeviceClockSelector, DeviceClockFrequency, DeviceTemperatureSelector, DeviceTemperature,
  LvDeviceUpTime, LvDeviceType, SensorWidth, SensorHeight,
  WidthMax, HeightMax, Width, Height,
  OffsetX, OffsetY, PixelFormat, BinningHorizontal,
  BinningVertical, DecimationHorizontal, DecimationVertical, LvAOIMode,
  LvReadoutWidth, LvReadoutHeight, LvReadoutOffsetX, LvReadoutOffsetY,
  LvVariablePayloadSize, AcquisitionMode, TriggerSelector, TriggerMode,
  TriggerSoftware, TriggerSource, TriggerActivation, TriggerDelay,
  TriggerDivider, LvTriggerCaching, ExposureMode, LvLongRangeExposureMode,
  LvGlobalResetMode, ExposureTime, ExposureAuto, LvAcquisitionFrameRateControlMode,
  AcquisitionFrameRate, LineSelector, LineMode, LineFormat,
  LineSource, LineInverter, LineStatus, LineStatusAll,
  UserOutputSelector, UserOutputValue, UserOutputValueAll, UserOutputValueAllMask,
  CounterSelector, LvCounterMode, CounterEventSource, CounterReset,
  CounterValue, CounterDuration, TimerSelector, TimerDuration,
  TimerDelay, TimerTriggerSource, LvSpecialPurposeTriggerSelector, LvSpecialPurposeTriggerSource,
  LvSpecialPurposeTriggerActivation, LvSpecialPurposeTriggerSoftware, LvImageStampsResetMask, Lv↩
  ImageStampSelector,
  LvImageStampResetEnable, LvBootSwitch, LvBayerDecoderAlgorithm, LvBayerDecoderThreshold,
  LvWatchdogEnable, LvWatchdogTimerDuration, LvWatchdogTimerReset, LvWatchdogFailed,
  GainSelector, Gain, GainAuto, BlackLevelSelector,
  BlackLevel, BlackLevelAuto, ColorTransformationSelector, ColorTransformationEnable,
  ColorTransformationValueSelector, ColorTransformationValue, LvExternalDeviceControlMode, LvExternal↩
  ADCSelector,
  LvExternalADCValue, LvPowerSwitchCurrentAction, LvPowerSwitchSelector, LvPowerSwitchBoundADC,
  LvPowerSwitchDrive, LvPowerSwitchPulsePlus, LvPowerSwitchPulseMinus, LvLensControlCalibrate,
  LvLensControlMinusEnd, LvLensControlPlusEnd, LvLensControlPulsePeriod, LvLensControlDutyCycle,
  LvLensControlTargetApproach, LvLensControlNrSlowSteps, LvLensControlTargetPosition, LvLensControl↩
  AdjustPosition,
  LvPowerSwitchPulseDuration, LvLensControlMinCalibrationRange, LvLensControlCalibrateAll, LUTSelector,
  LUTEnable, LUTIndex, LUTValue, LUTValueAll,
  PayloadSize, GevVersionMajor, GevVersionMinor, GevDeviceModeIsBigEndian,
  GevDeviceModeCharacterSet, GevInterfaceSelector, GevMACAddress, GevSupportedOptionSelector,
  GevSupportedOption, GevCurrentIPConfigurationLLA, GevCurrentIPConfigurationDHCP, GevCurrentIP↩
  ConfigurationPersistentIP,
  GevCurrentIPAddress, GevCurrentSubnetMask, GevCurrentDefaultGateway, GevPersistentIPAddress,
  GevPersistentSubnetMask, GevPersistentDefaultGateway, GevNumberOfInterfaces, GevMessageChannel↩
  Count,
  GevStreamChannelCount, GevHeartbeatTimeout, GevTimestampTickFrequency, GevTimestampControl↩

Latch,
GevTimestampControlReset, GevTimestampControlLatchReset, GevTimestampValue, GevCCP,
GevStreamChannelSelector, GevSCPInterfaceIndex, GevSCPHostPort, GevSCPSFireTestPacket,
GevSCPSDoNotFragment, GevSCPSBigEndian, GevSCPSPacketSize, GevSCPD,
GevSCDA, GevLinkSpeed, UserSetSelector, UserSetLoad,
UserSetSave, UserSetDefaultSelector, ChunkModeActive, ChunkSelector,
ChunkEnable, ChunkOffsetX, ChunkOffsetY, ChunkWidth,
ChunkHeight, ChunkPixelFormat, ChunkLinePitch, ChunkFrameID,
ChunkTimestamp, ChunkExposureTime, ChunkGainSelector, ChunkGain,
ChunkBlackLevel, ChunkLineStatusAll, ChunkLvExternalADCSelector, ChunkLvExternalADCValue,
EventSelector, EventNotification, LvSmartAppID, LvSmartAppInt1,
LvSmartAppInt2, LvSmartAppInt3, LvSmartAppInt4, LvSmartAppInt5,
LvSmartAppInt6, LvSmartAppInt7, LvSmartAppInt8, LvSmartAppInt9,
LvSmartAppInt10, LvSmartAppInt11, LvSmartAppInt12, LvSmartAppInt13,
LvSmartAppInt14, LvSmartAppInt15, LvSmartAppInt16, LvSmartAppInt17,
LvSmartAppInt18, LvSmartAppInt19, LvSmartAppInt20, LvSmartAppInt21,
LvSmartAppInt22, LvSmartAppInt23, LvSmartAppInt24, LvSmartAppInt25,
LvSmartAppInt26, LvSmartAppInt27, LvSmartAppInt28, LvSmartAppInt29,
LvSmartAppInt30, LvSmartAppInt31, LvSmartAppInt32, LvSmartAppUint1,
LvSmartAppUint2, LvSmartAppUint3, LvSmartAppUint4, LvSmartAppUint5,
LvSmartAppUint6, LvSmartAppUint7, LvSmartAppUint8, LvSmartAppUint9,
LvSmartAppUint10, LvSmartAppUint11, LvSmartAppUint12, LvSmartAppUint13,
LvSmartAppUint14, LvSmartAppUint15, LvSmartAppUint16, LvSmartAppUint17,
LvSmartAppUint18, LvSmartAppUint19, LvSmartAppUint20, LvSmartAppUint21,
LvSmartAppUint22, LvSmartAppUint23, LvSmartAppUint24, LvSmartAppUint25,
LvSmartAppUint26, LvSmartAppUint27, LvSmartAppUint28, LvSmartAppUint29,
LvSmartAppUint30, LvSmartAppUint31, LvSmartAppUint32, LvSmartAppAsciiCmdString,
LvSmartAppAsciiCmdExecute, LvSmartAppAsciiCmdFeedback, LvSmartAppAsciiCmdRetCode, LvSmart↩
AppPath,
LvSmartAppStart, EventLvLog, EventLvLogTimestamp, EventLvLogMessage,
EventLvSmartAppLog, EventLvSmartAppLogTimestamp, EventLvSmartAppLogMessage, LvSerialPort↩
BaudRate,
LvSerialPortParity, LvSerialPortDataBits, LvSerialPortStopBits, LvSerialPortTimeout,
LvSerialPortEOTMarker, LvSerialPortMaxResponseLength, LvSerialPortCommandString, LvSerialPort↩
CommandSend,
LvSerialPortCommandResponse, LvSerialPortCommandStatus, LvSmartAppExitEvent, LvWatchdogTimer↩
Value,
LvLensControlInvertedPolarity, GevMCPHostPort, GevMCDA, GevMCTT,
GevMCRC, ChunkLvSmartAppString, ChunkLvSmartAppIntSelector, ChunkLvSmartAppInt,
ChunkLvSmartAppUintSelector, ChunkLvSmartAppUint, ChunkLvSmartAppRegister, EventLvSmartApp↩
String,
EventLvSmartAppStringTimestamp, EventLvSmartAppStringValue, EventLvSmartAppInt, EventLvSmart↩
AppIntTimestamp,
EventLvSmartAppIntSelector, EventLvSmartAppIntValue, EventLvSmartAppUint, EventLvSmartAppUint↩
Timestamp,
EventLvSmartAppUintSelector, EventLvSmartAppUintValue, EventLvSmartAppRegister, EventLvSmart↩
AppRegisterTimestamp,
EventLvSmartAppRegisterValue, DeviceSFNCVersionMajor, DeviceSFNCVersionMinor, DeviceSFNC↩
VersionSubMinor,
LvLineDebounceDuration, ActionDeviceKey, ActionSelector, ActionGroupKey,
ActionGroupMask, LvLensControlCalibrationStatus, LvLUTMode, BalanceRatioSelector,
BalanceRatio, BalanceWhiteAuto, GevDeviceClass, GevIPConfigurationStatus,
GevDiscoveryAckDelay, GevGVCPExtendedStatusCodes, GevGVCPPendingAck, GevGVCPHeartbeat↩
Disable,
GevGVCPPendingTimeout, GevPrimaryApplicationSwitchoverKey, GevPrimaryApplicationSocket, Gev↩

PrimaryApplicationIPAddress,
GevMCSP, GevSCCFGUnconditionalStreaming, GevSCCFGExtendedChunkData, GevSCPDirection,
GevSCSP, ChunkLvTriggerDelayed, EventLvTriggerDropped, EventLvTriggerDroppedTimestamp,
LvStrobeEnable, LvStrobeDurationMode, LvStrobeDuration, LvStrobeDelay,
LvStrobeBrightness, LvStrobeDropMode, LvLUTReset, ChunkLvStrobeDropped,
ReverseX, ReverseY, RegionSelector, RegionMode,
RegionDestination, AcquisitionFrameCount, AcquisitionBurstFrameCount, LvCustomID,
LvCustomInfo, LvCustomRegMode, LvCustomRegAddr, LvCustomRegData,
LvCustomRegMux, LinePitch, ChunkLvFrameAbort, ChunkLvTriggerDropped,
ChunkLvTriggerError, ChunkLvEncoderPosition, ChunkLvEncoderRotation, DeviceID,
DeviceType, GevDeviceIPAddress, GevDeviceSubnetMask, GevDeviceMACAddress,
GevDeviceGateway, LvGevDeviceStreamCaptureMode, StreamSelector, StreamID,
DeviceEndianessMechanism, LvGevFindMaxPacketSize, LvGevPacketSizeValue, LvGevTestPacketSize,
LvGevPacketSizeTestSuccess, LvGevCCTT, LvGevCCRC, LvCCStatus,
LvDeviceDisplayName, LvDeviceIsAcquiring, LvUniProcessMode, LvUniProcessEnableInPlace,
LvUniPixelFormat, LvUniProcessPayloadSize, LvUniLinePitch, LvUniBayerDecoderAlgorithm,
LvUniBrightness, LvUniContrast, LvUniGamma, LvUniBalanceRatioSelector,
LvUniBalanceRatio, LvUniBalanceWhiteAuto, LvUniBalanceWhiteReset, LvUniColorTransformationSelector,
LvUniColorTransformationEnable, LvUniColorTransformationValueSelector, LvUniColorTransformationValue,
LvUniSaturation,
LvUniProcessExecution, LvUniLUTMode, LvUniLUTSelector, LvUniLUTEnable,
LvUniLUTIndex, LvUniLUTValue, LvUniLUTValueAll, LvUniColorTransformationMode,
Info }

- enum LvStreamFtr : LvFeature {
  StreamID, StreamAnnouncedBufferCount, StreamAcquisitionModeSelector, StreamAnnounceBuffer↩
  Minimum,
  StreamType, LvStreamDisplayName, LvCalcPayloadSize, LvPostponeQueueBuffers,
  LvAwaitDeliveryLimit, LvAutoAllocateProcessBuffers, LvPreallocateProcessBuffers, LvNumDelivered,
  LvNumUnderrun, LvNumAnnounced, LvNumQueued, LvNumAwaitDelivery,
  LvIsGrabbing, LvNumAborted, LvNumStarted, Info }
- enum LvRendererFtr : LvFeature {
  LvAutoDisplay, LvRenderType, LvOffsetX, LvOffsetY,
  LvWidth, LvHeight, LvIgnoreAspectRatio, LvDisableScaleUp,
  LvDisableScaleDown, LvCenterImage, LvNumberOfTiles, LvColumns,
  LvRows, LvTileGap, LvAutoTileCalculation, Info }
- enum LvEventFtr : LvFeature { EventType, NumInQueue, NumFired }
- enum LvBufferFtr : LvFeature {
  Base, Size, UserPtr, TimeStamp,
  NewData, IsQueued, IsAcquiring, IsIncomplete,
  TIType, SizeFilled, Width, Height,
  XOffset, YOffset, XPadding, YPadding,
  FrameId, ImagePresent, ImageOffset, PayloadType,
  PixelFormat, PixelFormatNameSpace, DeliveredImageHeight, DeliveredChunkPayloadSize,
  ChunkLayoutId, FileName, UniBase, UniSize,
  ProcessBase, ProcessSize, ExecProcess, UniImageOffset }
- enum LvInfoDataType : LvEnum {
  Unknown, String, StringList, Int16,
  UInt16, Int32, UInt32, Int64,
  UInt64, Float64, Ptr, Bool,
  SizeT, Buffer }
- enum LvQueueOperation : LvEnum {
  InputToOutput, OutputDiscard, AllToInput, UnqueuedToInput,
  AllDiscard }
- enum LvEventType : LvEnum { Error, NewBuffer, FeatureDevEvent }
- enum LvEventDataInfo : LvEnum { Id, Value }
- enum LvPixelFormat : LvEnum {

Mono8 = 0x01080001, Mono8S = 0x01080002, Mono10 = 0x01100003, Mono10Packed = 0x010C0004,
Mono12 = 0x01100005, Mono12Packed = 0x010C0006, Mono14 = 0x01100025, Mono16 = 0x01100007,
BayerGR8 = 0x01080008, BayerRG8 = 0x01080009, BayerGB8 = 0x0108000A, BayerBG8 = 0x0108000B,
BayerGR10 = 0x0110000C, BayerRG10 = 0x0110000D, BayerGB10 = 0x0110000E, BayerBG10 = 0x0110000F,
BayerGR12 = 0x01100010, BayerRG12 = 0x01100011, BayerGB12 = 0x01100012, BayerBG12 = 0x01100013,
BayerGR10Packed = 0x010C0026, BayerRG10Packed = 0x010C0027, BayerGB10Packed = 0x010C0028,
BayerBG10Packed = 0x010C0029,
BayerGR12Packed = 0x010C002A, BayerRG12Packed = 0x010C002B, BayerGB12Packed = 0x010C002C,
BayerBG12Packed = 0x010C002D,
BayerGR16 = 0x0110002E, BayerRG16 = 0x0110002F, BayerGB16 = 0x01100030, BayerBG16 = 0x01100031,
RGB8 = 0x02180014, BGR8 = 0x02180015, RGBA8 = 0x02200016, BGRA8 = 0x02200017,
RGB10 = 0x02300018, BGR10 = 0x02300019, RGB12 = 0x0230001A, BGR12 = 0x0230001B,
RGB16 = 0x02300033, RGB10V1Packed = 0x0220001C, RGB10P32 = 0x0220001D, RGB12V1Packed = 0x02240034,
RGB565P = 0x02100035, BGR565P = 0x02100036, YUV411_8 = 0x020C001E, YUV422_8_UYVY = 0x0210001F,
YUV422_8 = 0x02100032, YUV8 = 0x02180020, YCbCr422_8 = 0x0210003B, YCbCr601_422_8 = 0x0210003E,
YCbCr601_422_8_CbYCrY = 0x02100044, RGB8_Planar = 0x02180021, RGB10_Planar = 0x02300022,
RGB12_Planar = 0x02300023,
RGB16_Planar = 0x02300024, BGR555P = 0x021000E1, Mono8Signed, RGB8Packed,
BGR8Packed, RGBA8Packed, BGRA8Packed, RGB10Packed,
BGR10Packed, RGB12Packed, BGR12Packed, RGB16Packed,
RGB10V2Packed, RGB565Packed, BGR565Packed, YUV411Packed,
YUV422Packed, YUV422YUYVPacked, YUV444Packed, RGB8Planar,
RGB10Planar, RGB12Planar, RGB16Planar, Mono8s,
RGBa8, BGRa8, RGB565p, BGR565p,
RGB10p32, BGR555p, YUV411_8_UYYVYY, YUV8_UYV }

- enum LvDeviceAccess : LvEnum { None = 1, ReadOnly = 2, Control = 3, Exclusive = 4 }
- enum LvDeviceAccessStatus : LvEnum { Unknown = 0, ReadWrite = 1, ReadOnly = 2, NoAccess = 3 }
- enum LvDeviceScanType : LvEnum { Areascan, Linescan }
- enum LvDeviceClockSelector : LvEnum { SensorDigitization }
- enum LvDeviceTemperatureSelector : LvEnum { Sensor, Mainboard }
- enum LvAOIMode : LvEnum { Automatic, ClipOnTransfer, Manual }
- enum LvAcquisitionMode : LvEnum { SingleFrame, MultiFrame, Continuous }
- enum LvExposureAuto : LvEnum { Off, Once, Continuous }
- enum LvTriggerSelector : LvEnum { FrameStart, FrameBurstStart }
- enum LvTriggerMode : LvEnum { Off, On }
- enum LvTriggerSource : LvEnum {
  Line1, Line2, Line3, Line4,
  Line5, Line6, Line7, Line8,
  Line17, Line18, Line19, Line20,
  Line21, Line22, Line23, Line24,
  Software, Action1, Action2, Action3,
  Action4, Action5, Action6, Action7,
  Action8, Quad, Counter1, Counter2,
  Counter3, Counter4, Timer1, Timer2,
  Timer3, Timer4 }
- enum LvTriggerActivation : LvEnum { RisingEdge, FallingEdge }
- enum LvTriggerCaching : LvEnum { Cache, Drop }
- enum LvExposureMode : LvEnum { Timed }
- enum LvAcquisitionFrameRateControlMode : LvEnum { Off, On }

- enum LvLineSelector : LvEnum {
  Line1, Line2, Line3, Line4,
  Line5, Line6, Line7, Line8,
  Line9, Line10, Line11, Line12,
  Line13, Line14, Line15, Line16,
  Line17, Line18, Line19, Line20,
  Line21, Line22, Line23, Line24,
  Line25, Line26, Line27, Line28,
  Line29, Line30, Line31, Line32 }
- enum LvLineMode : LvEnum { Input, Output }
- enum LvLineFormat : LvEnum {
  NoConnect, TriState, TTL, LVDS,
  RS422, OptoCoupled }
- enum LvLineSource : LvEnum {
  Off, ExposureActive, Timer1Active, Timer2Active,
  Timer3Active, Timer4Active, UserOutput1, UserOutput2,
  UserOutput3, UserOutput4, UserOutput5, UserOutput6,
  UserOutput7, UserOutput8, Counter1, Counter2,
  Counter3, Counter4 }
- enum LvCounterSelector : LvEnum { Counter1, Counter2, Counter3, Counter4 }
- enum LvCounterMode : LvEnum { Autoreset }
- enum LvCounterEventSource : LvEnum {
  Off, FrameTrigger, TimerTick, Line1,
  Line2, Line3, Line4, Line17,
  Line18 }
- enum LvTimerSelector : LvEnum { Timer1, Timer2, Timer3, Timer4 }
- enum LvTimerTriggerSource : LvEnum {
  Off, FrameTrigger, Counter1End, Counter2End,
  Counter3End, Counter4End, UserOutput1, UserOutput2,
  UserOutput3, UserOutput4, UserOutput5, UserOutput6,
  UserOutput7, UserOutput8 }
- enum LvSpecialPurposeTriggerSelector : LvEnum { ImageStampsReset }
- enum LvSpecialPurposeTriggerSource : LvEnum {
  Off, Line1, Line2, Line3,
  Line4, Line5, Line6, Line7,
  Line8, Line17, Line18, Line19,
  Line20, Line21, Line22, Line23,
  Line24, Action1, Action2, Action3,
  Action4, Action5, Action6, Action7,
  Action8 }
- enum LvSpecialPurposeTriggerActivation : LvEnum { RisingEdge, FallingEdge }
- enum LvImageStampSelector : LvEnum { Timestamp, FrameID }
- enum LvBootSwitch : LvEnum { PureGEV, Legacy }
- enum LvGainSelector : LvEnum { All, AnalogAll, DigitalAll }
- enum LvGainAuto : LvEnum { Off, Once, Continuous }
- enum LvBlackLevelSelector : LvEnum { All }
- enum LvBlackLevelAuto : LvEnum { Off, Once, Continuous }
- enum LvColorTransformationSelector : LvEnum { RGBtoRGB }
- enum LvColorTransformationValueSelector : LvEnum {
  Gain00, Gain01, Gain02, Gain10,
  Gain11, Gain12, Gain20, Gain21,
  Gain22 }
- enum LvExternalDeviceControlMode : LvEnum { Custom }
- enum LvExternalADCSelector : LvEnum { ExternalADC1, ExternalADC2, ExternalADC3, ExternalADC4 }
- enum LvPowerSwitchCurrentAction : LvEnum {
  Idle, Pulse, Calibrate, AdjustPosition,
  Drive }

- enum LvPowerSwitchSelector : LvEnum { PowerSwitch1, PowerSwitch2, PowerSwitch3, PowerSwitch4 }
- enum LvPowerSwitchDrive : LvEnum { Off, Plus, Minus }
- enum LvPowerSwitchDriveAll : LvEnum { Off, Plus, Minus }
- enum LvPowerSwitchBoundADC : LvEnum { None, ExternalADC1, ExternalADC2, ExternalADC3, ExternalADC4 }
- enum LvLensControlTargetApproach : LvEnum { Direct, FromPlus, FromMinus }
- enum LvLUTSelector : LvEnum { Luminance, Red, Green, Blue }
- enum LvGevDeviceModeCharacterSet : LvEnum { UTF8 }
- enum LvGevSupportedOptionSelector : LvEnum { IPConfigurationLLA, IPConfigurationDHCP, IPConfigurationPersistentIP, CommandsConcatenation, WriteMem, PacketResend, Event, EventData, PendingAck, Action, PrimaryApplicationSwitchover, ExtendedStatusCodes, DiscoveryAckDelayWritable, DiscoveryAckDelay, TestData, ManifestTable, CCPApplicationSocket, LinkSpeed, HeartbeatDisable, SerialNumber, UserDefinedName, StreamChannelSourceSocket, StreamChannel0ExtendedChunkData, StreamChannel0UnconditionalStreaming, StreamChannel0IPReassembly, StreamChannel0BigAndLittleEndian, MessageChannelSourceSocket }
- enum LvGevCCP : LvEnum { OpenAccess, ExclusiveAccess, ControlAccess, ControlAccessSwitchoverActive }
- enum LvUserSetSelector : LvEnum { Default, UserSet1, UserSet2, UserSet3, UserSet4 }
- enum LvUserSetDefaultSelector : LvEnum { Default, UserSet1, UserSet2, UserSet3, UserSet4, None }
- enum LvChunkSelector : LvEnum { OffsetX, OffsetY, Width, Height, PixelFormat, LinePitch, FrameID, Timestamp, ExposureTime, Gain, LineStatusAll, BlackLevel, LvExternalADCValue, LvSmartAppString, LvSmartAppInt, LvSmartAppUint, LvSmartAppRegister, LvTriggerDelayed, LvStrobeDropped, LvFrameAbort, LvTriggerDropped, LvTriggerError, LvEncoderPosition, LvEncoderRotation }
- enum LvChunkGainSelector : LvEnum { AnalogAll, DigitalAll }
- enum LvEventSelector : LvEnum { LvLog, LvSmartAppLog, LvSmartAppString, LvSmartAppInt, LvSmartAppUint, LvSmartAppRegister, LvTriggerDropped }
- enum LvEventNotification : LvEnum { Off, On }
- enum LvTLType : LvEnum { Mixed, Custom, GEV }
- enum LvInterfaceType : LvEnum { Custom, GEV }
- enum LvDeviceType : LvEnum { Custom, GEV }
- enum LvGevDeviceStreamCaptureMode : LvEnum { SystemDefault, Socket, FilterDriver }
- enum LvStreamAcquisitionModeSelector : LvEnum { Default }
- enum LvStreamType : LvEnum { Custom, GEV }
- enum LvUniProcessMode : LvEnum { HwOnly, SwOnly, Auto, Off }
- enum LvBayerDecoderAlgorithm : LvEnum { NearestNeighbour, BilinearInterpolation, BilinearColorCorrection, PixelGrouping, VariableGradient }
- enum LvUniBalanceRatioSelector : LvEnum { Red, Green, Blue }
- enum LvUniBalanceWhiteAuto : LvEnum { Off, Once }
- enum LvUniColorTransformationSelector : LvEnum { RGBtoRGB }
- enum LvUniColorTransformationValueSelector : LvEnum { Gain00, Gain01, Gain02, Gain10, Gain11, Gain12, Gain20, Gain21, Gain22 }
- enum LvRenderType : LvEnum { FullSize, ScaleToFit, ScaleToSize, ScaleToTiles }

- enum LvRenderFlags : UInt32 { None, RepaintBackground, DontPaintIncomplete, IgnoreInvalidWinHandle }
- enum LvFindBy : LvEnum {
  UserID, VendorName, ModelName, TLType,
  DisplayName, GevIPAddress, GevMACAddress, SerialNumber,
  Any }
- enum LvSerialPortBaudRate : LvEnum {
  Baud2400, Baud4800, Baud9600, Baud14400,
  Baud19200, Baud38400, Baud57600, Baud115200 }
- enum LvSerialPortParity : LvEnum { None, Odd, Even }
- enum LvSerialPortDataBits : LvEnum { DataBits7, DataBits8 }
- enum LvSerialPortStopBits : LvEnum { StopBits1, StopBits1dot5, StopBits2 }
- enum LvSerialPortCommandStatus : LvEnum {
  Success, Timeout, PortBusy, CommunicationError,
  FrameError, ParityError, Overflow }
- enum LvChunkLvExternalADCSelector : LvEnum { ExternalADC1, ExternalADC2, ExternalADC3, External↩ ADC4 }
- enum LvUserOutputSelector : LvEnum {
  UserOutput1, UserOutput2, UserOutput3, UserOutput4,
  UserOutput5, UserOutput6, UserOutput7, UserOutput8 }
- enum LvUniProcessExecution : LvEnum { OnBufferPtrQuery, OnPopFromQueue, OnExplicitRequest }
- enum LvLensControlCalibrationStatus : LvEnum { Invalid, Valid }
- enum LvLUTMode : LvEnum { Direct, BalanceWhite }
- enum LvBalanceRatioSelector : LvEnum { Red, Green, Blue }
- enum LvBalanceWhiteAuto : LvEnum { Off, Once, Continuous }
- enum LvGevDeviceClass : LvEnum { Transmitter }
- enum LvGevIPConfigurationStatus : LvEnum {
  None, PersistentIP, DHCP, LLA,
  ForceIP }
- enum LvGevSCPDirection : LvEnum { Transmitter }
- enum LvDeviceEndianessMechanism : LvEnum { Legacy, Standard }
- enum LvUniLUTMode : LvEnum { Direct, Generated }
- enum LvUniLUTSelector : LvEnum { Luminance, Red, Green, Blue }
- enum LvUniColorTransformationMode : LvEnum { Direct, Generated }
- enum LvStrobeEnable : LvEnum { Off, AllClusters, LEDCluster1, LEDCluster2 }
- enum LvStrobeDurationMode : LvEnum { FrameRateRelated, Free }
- enum LvStrobeDropMode : LvEnum { DropStrobe, DelayFrame }
- enum LvRegionSelector : LvEnum { Region0, Region1, Region2, Region3 }
- enum LvipImgAttr : UInt32 {
  None, BottomUp, DWordAligned, QWordAligned,
  SSEAligned, NotDataOwner }
- enum LvipOption : UInt32 {
  None, ReallocateDst, TiffConvertTo16Bit, BmpForceTopDown,
  BmpForceBottomUp, JpegConvertToBgr, JpegReadHeaderOnly, WbCorrectFactors }
- enum LvipLutType : LvEnum {
  Uni, Type8Bit, Type10Bit, Type12Bit,
  UniBayer, Type8BitBayer, Type10BitBayer, Type12BitBayer,
  UniBayer16, Type10BitBayer16, Type12BitBayer16 }
- enum LvipColor : LvEnum { None }
- enum LvipTextAttr : UInt32 {
  None, Bold, Italic, Underline,
  Strikeout, Nonantialiased, Shadow, Outline,
  ShadowRB, ShadowRT, ShadowLB, ShadowLT,
  ShadowB, ShadowT, ShadowR, ShadowL }

Functions

- private delegate void LvEventCallbackFunct (IntPtr pBuffer, SizeT Size, IntPtr pUserParam)
- private delegate void LvEventCallbackNewBufferFunct (LvHBuffer hBuffer, IntPtr pUserPointer, IntPtr pUser↩ Param)
- private delegate void LvFeatureCallbackFunct (IntPtr pUserParam, IntPtr pFeatureParam, String^Name)
- public delegate void LvEventHandler (Object^Sender, LvEventArgs^E)
- public delegate void LvEventNewBufferHandler (Object^Sender, LvNewBufferEventArgs^E)
- public delegate void LvFeatureChangedHandler (Object^Sender, LvFeatureChangedArgs^E)

7.2.1 Detailed Description

The NewElectronicTechnology::SynView namespace This namespace is the root namespace for the SynView .Net Class library.

7.2.2 Typedef Documentation

7.2.2.1 typedef LvHModule LvHBuffer

Typedef for a handle to the Buffer module. Used only internally in the lv.SynView.net.

7.2.2.2 typedef LvHModule LvHDevice

Typedef for a handle to the Device module. Used only internally in the lv.SynView.net.

7.2.2.3 typedef LvHModule LvHEvent

Typedef for a handle to the Event module. Used only internally in the lv.SynView.net.

7.2.2.4 typedef LvHModule LvHInterface

Typedef for a handle to the Interface module. Used only internally in the lv.SynView.net.

7.2.2.5 typedef UInt32 LvHModule

Base typedef for a handle to a module. Used only internally in the lv.SynView.net.

7.2.2.6 typedef LvHModule LvHRenderer

Typedef for a handle to the Renderer module. Used only internally in the lv.SynView.net.

7.2.2.7 typedef LvHModule LvHStream

Typedef for a handle to the Stream module. Used only internally in the lv.SynView.net.

7.2.2.8 typedef LvHModule LvHSystem

Typedef for a handle to the System module. Used only internally in the lv.SynView.net.

7.2.3 Function Documentation

7.2.3.1 private delegate void NewElectronicTechnology::SynView::LvEventCallbackFunct (IntPtr *pBuffer,* SizeT *Size,* IntPtr *pUserParam*)

Prototype for the LvEvent callback function. This declaration is used only internally in the LvEvent class, but as it must be declared as public, it is visible for the users. Do not use it, instead, assign to the LvEvent::OnEvent property the LvEventHandler() instance.

Parameters

pBuffer	Pointer to buffer, extracted from the output queue.
Size	Buffer size.
pUserParam	User parameter, supplied in the LvEvent::SetCallback() function. It enables the application to distingush from which object the callback was called in case the same callback function is shared by multiple Event modules.

7.2.3.2 private delegate void NewElectronicTechnology::SynView::LvEventCallbackNewBufferFunct (LvHBuffer *hBuffer,* IntPtr *pUserPointer,* IntPtr *pUserParam*)

Prototype for the LvEventNewBuffer callback function. This declaration is used only internally in the LvEvent class, but as it must be declared as public, it is visible for the users. Do not use it, instead, assign to the LvEvent::OnEventNewBuffer property the LvEventNewBufferHandler() instance.

Parameters

hBuffer	Handle to LvBuffer, extracted from the output queue.
pUserPointer	The pUserPointer of the LvBuffer is passed here. In the C++ wrapper class this is used to give direct pointer to the LvBuffer class instance.
pUserParam	User parameter, supplied in the LvEvent::SetCallbackNewBuffer() function. It enables the application to distingush from which object the callback was called in case the same callback function is shared by multiple Event modules.

7.2.3.3 public delegate void NewElectronicTechnology::SynView::LvEventHandler (Object^ *Sender,* LvEventArgs^ *E*)

The delegate prototype, which is to be used with the LvEvent::OnEvent handler.

Parameters

Sender	Sender of the event.
E	Event arguments, see LvEventArgs.

7.2.3.4 public delegate void NewElectronicTechnology::SynView::LvEventNewBufferHandler (Object^ *Sender,* LvNewBufferEventArgs^ *E*)

The delegate prototype, which is to be used with the LvEvent::OnEventNewBuffer handler.

Parameters

Sender	Sender of the event.
E	Event arguments, see LvNewBufferEventArgs.

7.2.3.5 private delegate void NewElectronicTechnology::SynView::LvFeatureCallbackFunct (IntPtr *pUserParam,* IntPtr *pFeatureParam,* String^ *Name*)

Prototype for the feature updated callback function. This declaration is used only internally in the LvModule class, but as it must be declared as public, it is visible for the users. Do not use it, instead, assign to the OnFeatureChanged property the LvFeatureChangedHandler() instance.

Parameters

pUserParam	User pUserParam, supplied in the LvSystem::RegisterFeatureCallback() function. It can be used to distinguish distingush from which object the callback was called in case the same callback function is shared by multiple Event modules.
pFeatureParam	The pFeatureParam passed in the LvSystem::RegisterFeatureCallback(). It is usually used to identify the feature, which has changed.
Name	The string ID of the feature.

7.2.3.6 public delegate void NewElectronicTechnology::SynView::LvFeatureChangedHandler (Object^ *Sender,* LvFeatureChangedArgs^ *E*)

The delegate prototype, which is to be used with the OnFeatureChanged handler.

Parameters

Sender	Sender of the event.
E	Event arguments, see LvFeatureChangedArgs.

Chapter 8

Class Documentation

8.1 abstract Class Reference

Public Member Functions

- void **set** (Boolean bEnable)
- LvStatus GetNumFeatures (LvFtrGroup FtrGroup, UInt32%NumFeatures)
- LvStatus StartPollingThread (UInt32 PollingTime)
- LvStatus StartPollingThread (UInt32 PollingTime, Boolean PollChildren)
- LvStatus StopPollingThread ()
- LvStatus Poll ()

Static Public Member Functions

- static UInt32 GetVersion ()
- static LvStatus OpenLibrary ()
- static LvStatus CloseLibrary ()
- static String GetErrorMessage (LvStatus Error)
- static String GetLastErrorMessage ()
- static void Log (String^LogMessage)
- static LvStatus GetLibInfo (LvLibInfo LibInfo, Int32%Info, Int32 Param)
- static LvStatus GetLibInfo (LvLibInfo LibInfo, Int32%Info)
- static LvStatus GetLibInfoStr (LvLibInfo LibInfo, String^%InfoStr)
- static LvStatus GetLibInfoStr (LvLibInfo LibInfo, String^%InfoStr, Int32 Param)
- static LvStatus UpdateSystemList ()
- static LvStatus GetNumberOfSystems (UInt32%NumberOfSystems)
- static LvStatus GetSystemId (UInt32 Index, String^%SystemId)

Public Attributes

- event LvFeatureChangedHandler OnFeatureChanged

Static Public Attributes

- static property Boolean ThrowErrorEnable

Protected Member Functions

- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvFeature%Feature)
- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvFeature%Feature, UInt32%Level)
- LvStatus GetFeatureByName (LvFtrGroup FtrGroup, String^Name, LvFeature%Feature)
- Boolean IsImplemented (LvFeature Feature)
- Boolean IsImplementedByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsAvailable (LvFeature Feature)
- Boolean IsAvailableByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsReadable (LvFeature Feature)
- Boolean IsWritable (LvFeature Feature)
- Boolean IsAvailableEnumEntry (LvFeature Feature, LvEnum EnumEntry)
- Boolean IsImplementedEnumEntry (LvFeature Feature, LvEnum EnumEntry)
- LvStatus GetType (LvFeature Feature, LvFtrType%FtrType)
- LvStatus GetType (LvFeature Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup)
- LvStatus GetBool (LvFeature Feature, Boolean%Value)
- LvStatus SetBool (LvFeature Feature, Boolean Value)
- LvStatus GetInt32 (LvFeature Feature, Int32%Value)
- LvStatus SetInt32 (LvFeature Feature, Int32 Value)
- LvStatus GetInt32Range (LvFeature Feature, Int32%MinValue, Int32%MaxValue)
- LvStatus GetInt32Range (LvFeature Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment)
- LvStatus GetInt64 (LvFeature Feature, Int64%Value)
- LvStatus SetInt64 (LvFeature Feature, Int64 Value)
- LvStatus GetInt64Range (LvFeature Feature, Int64%MinValue, Int64%MaxValue)
- LvStatus GetInt64Range (LvFeature Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment)
- LvStatus GetInt (LvFeature Feature, Int64%Value)
- LvStatus SetInt (LvFeature Feature, Int64 Value)
- LvStatus GetIntRange (LvFeature Feature, Int64%MinValue, Int64%MaxValue)
- LvStatus GetIntRange (LvFeature Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment)
- LvStatus GetFloat (LvFeature Feature, Double%Value)
- LvStatus SetFloat (LvFeature Feature, Double Value)
- LvStatus GetFloatRange (LvFeature Feature, Double%MinValue, Double%MaxValue)
- LvStatus GetFloatRange (LvFeature Feature, Double%MinValue, Double%MaxValue, Double%Increment)
- LvStatus GetString (LvFeature Feature, String^%Value)
- LvStatus SetString (LvFeature Feature, String^Value)
- LvStatus GetBuffer (LvFeature Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetBufferSize (LvFeature Feature, SizeT%Size)
- LvStatus SetBuffer (LvFeature Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetPtr (LvFeature Feature, IntPtr%pValue)
- LvStatus SetPtr (LvFeature Feature, IntPtr pValue)
- LvStatus GetEnum (LvFeature Feature, LvEnum%Value)
- LvStatus SetEnum (LvFeature Feature, LvEnum Value)
- LvStatus GetEnumStr (LvFeature Feature, String^%SymbolicName)
- LvStatus SetEnumStr (LvFeature Feature, String^SymbolicName)
- LvStatus GetEnumValByStr (LvFeature Feature, String^SymbolicName, LvEnum%Value)
- LvStatus GetEnumValByStr (LvFeature Feature, String^SymbolicName, LvEnum%Value, LvFtrAccess%Ftr↩ Access)
- LvStatus GetEnumStrByVal (LvFeature Feature, LvEnum Value, String^%SymbolicName)
- LvStatus GetEnumStrByVal (LvFeature Feature, LvEnum Value, String^%SymbolicName, LvFtr↩ Access%FtrAccess)
- LvStatus CmdExecute (LvFeature Feature)
- LvStatus CmdExecute (LvFeature Feature, UInt32 Timeout)
- LvStatus CmdIsDone (LvFeature Feature, Boolean%IsDone)
- LvStatus GetAccess (LvFeature Feature, LvFtrAccess%FtrAccess)
- LvStatus GetVisibility (LvFeature Feature, LvFtrVisibility%FtrVisibility)

- LvStatus GetInfo (LvFeature Feature, LvFtrInfo FtrInfo, Int32%Info)
- LvStatus GetInfo (LvFeature Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param)
- LvStatus GetInfoStr (LvFeature Feature, LvFtrInfo FtrInfo, String^%InfoStr)
- LvStatus GetInfoStr (LvFeature Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param)
- LvStatus RegisterFeatureCallback (LvFeature Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩ FeatureParam)
- **LvModule** ()
- void **CallbackFeatureChanged** (IntPtr pUserParam, IntPtr pFeatureParam, String^Name)

Protected Attributes

- internal **__pad0__**: LvFeatureChangedArgs^ m_pFeatureChangedArgs
- LvFeatureCallbackFunct **m_pLvFeatureCallbackFunct**
- LvHModule **m_hModule**

8.1.1 Detailed Description

The base class for all modules. It provides methods for manipulating the features, if the module provides any. This class cannot be instantiated, it only serves as a base class.

8.1.2 Member Data Documentation

8.1.2.1 event LvFeatureChangedHandler OnFeatureChanged

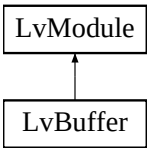
The event handler, which is called whenever a feature is changed. The handler is equivalent to a callback function in the DLL version.

Important:

SynView calls the event functions from a separate thread, i.e. the function can be called asynchronously, while your application is executing its code somewhere else. Be aware of this.

8.2 LvBuffer Class Reference

Inheritance diagram for LvBuffer:



Public Member Functions

- LvStatus AttachProcessBuffer (IntPtr pDataPointer, SizeT DataSize)
- LvStatus Queue ()
- LvStatus ParseChunkData ()
- LvStatus ParseChunkData (Boolean UpdateLayout)
- LvStatus SaveImageToBmpFile (String^FileName)
- LvStatus SaveImageToJpgFile (String^FileName, UInt32 Quality)
- LvStatus SaveImageToTifFile (String^FileName)

- LvStatus SaveImageToTifFile (String^ FileName, UInt32 Options)
- LvStatus GetLvipImage (LvipImage^ %Image)
- LvStatus GetLastPaintRect (Int32%X, Int32%Y, Int32%Width, Int32%Height)
- LvStatus UniCalculateWhiteBalance ()
- LvHBuffer GetHandle ()
- IntPtr GetUserPtr ()
- LvStream GetParentStream ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvBufferFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvBufferFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^ Name, LvBufferFtr%Feature)
- Boolean **IsImplemented** (LvBufferFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^ Name)
- Boolean **IsAvailable** (LvBufferFtr Feature)
- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^ Name)
- Boolean **IsReadable** (LvBufferFtr Feature)
- Boolean **IsWritable** (LvBufferFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvBufferFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvBufferFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvBufferFtr Feature, LvFtrType%FtrType) new
- LvStatus **GetType** (LvBufferFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvBufferFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvBufferFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvBufferFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvBufferFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvBufferFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvBufferFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvBufferFtr Feature, Int64%Value)
- LvStatus **SetInt64** (LvBufferFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvBufferFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvBufferFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvBufferFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvBufferFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvBufferFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvBufferFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvBufferFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvBufferFtr Feature, Double Value)
- LvStatus **GetFloatRange** (LvBufferFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvBufferFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvBufferFtr Feature, String^ %Value)
- LvStatus **SetString** (LvBufferFtr Feature, String^ Value)
- LvStatus **GetBuffer** (LvBufferFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvBufferFtr Feature, SizeT%Size)
- LvStatus **SetBuffer** (LvBufferFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvBufferFtr Feature, IntPtr%pValue)
- LvStatus **SetPtr** (LvBufferFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvBufferFtr Feature, LvEnum%Value)
- LvStatus **SetEnum** (LvBufferFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvBufferFtr Feature, String^ %SymbolicName)
- LvStatus **SetEnumStr** (LvBufferFtr Feature, String^ SymbolicName)
- LvStatus **GetEnumValByStr** (LvBufferFtr Feature, String^ SymbolicName, LvEnum%Value) new
- LvStatus **GetEnumValByStr** (LvBufferFtr Feature, String^ SymbolicName, LvEnum%Value, LvFtr↩ Access%FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvBufferFtr Feature, LvEnum Value, String^ %SymbolicName) new

- LvStatus **GetEnumStrByVal** (LvBufferFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩ Access%FtrAccess) new
- LvStatus **CmdExecute** (LvBufferFtr Feature) new
- LvStatus **CmdExecute** (LvBufferFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvBufferFtr Feature, Boolean%IsDone)
- LvStatus **GetAccess** (LvBufferFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus **GetVisibility** (LvBufferFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus **GetInfo** (LvBufferFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus **GetInfo** (LvBufferFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvBufferFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus **GetInfoStr** (LvBufferFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus **RegisterFeatureCallback** (LvBufferFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩ FeatureParam)

Static Public Member Functions

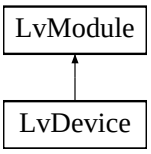
- static LvStatus Open (LvStream^Stream, IntPtr pDataPointer, SizeT DataSize, IntPtr pUserPointer, UInt32 Options, LvBuffer^%Buffer)
- static LvStatus Close (LvBuffer^%Buffer)

8.2.1 Detailed Description

The LvBuffer class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/↩ Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.3 LvDevice Class Reference

Inheritance diagram for LvDevice:



Public Member Functions

- LvStatus GetNumberOfStreams (UInt32%NumberOfStreams)
- LvStatus GetStreamId (UInt32 Index, String^%StreamId)
- LvStatus AcquisitionStart ()
- LvStatus AcquisitionStart (UInt32 Options)
- LvStatus AcquisitionStop ()
- LvStatus AcquisitionStop (UInt32 Options)
- LvStatus AcquisitionAbort ()
- LvStatus AcquisitionAbort (UInt32 Options)
- LvStatus AcquisitionArm ()
- LvStatus AcquisitionArm (UInt32 Options)
- LvStatus SaveSettings (String^Id, String^FileName, UInt32 Options)

- LvStatus LoadSettings (String^ Id, String^ FileName, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< Byte >^ ByteArray, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< Byte >^ ByteArray, SizeT Size, UInt32 Options)
- LvStatus UniSetLut (LvLUTSelector Selector, array< UInt16 >^ UInt16Array, SizeT Size)
- LvStatus UniSetLut (LvLUTSelector Selector, array< UInt16 >^ UInt16Array, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, IntPtr pLUT, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< Byte >^ ByteArray, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< Byte >^ ByteArray, SizeT Size, UInt32 Options)
- LvStatus UniGetLut (LvLUTSelector Selector, array< UInt16 >^ UInt16Array, SizeT Size)
- LvStatus UniGetLut (LvLUTSelector Selector, array< UInt16 >^ UInt16Array, SizeT Size, UInt32 Options)
- LvStatus FwGetFilePattern (UInt32 Which, String^ %FilePattern)
- LvStatus FwLoad (UInt32 Which, String^ FilePath)
- LvStatus FwGetLoadStatus (UInt32 Which, UInt32%CurrentByteCount, Boolean%IsLoading)
- LvStatus OpenStream (String^ StreamId, LvStream^ %Stream)
- LvStatus CloseStream (LvStream^ %Stream)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^ %Event)
- LvStatus CloseEvent (LvEvent^ %Event)
- LvHDevice GetHandle ()
- LvInterface GetParentInterface ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvDeviceFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvDeviceFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^ Name, LvDeviceFtr%Feature)
- Boolean **IsImplemented** (LvDeviceFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^ Name)
- Boolean **IsAvailable** (LvDeviceFtr Feature)
- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^ Name)
- Boolean **IsReadable** (LvDeviceFtr Feature)
- Boolean **IsWritable** (LvDeviceFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvDeviceFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvDeviceFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvDeviceFtr Feature, LvFtrType%FtrType) new
- LvStatus **GetType** (LvDeviceFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvDeviceFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvDeviceFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvDeviceFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvDeviceFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvDeviceFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvDeviceFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvDeviceFtr Feature, Int64%Value)
- LvStatus **SetInt64** (LvDeviceFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvDeviceFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvDeviceFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvDeviceFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvDeviceFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvDeviceFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvDeviceFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvDeviceFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvDeviceFtr Feature, Double Value)
- LvStatus **GetFloatRange** (LvDeviceFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvDeviceFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvDeviceFtr Feature, String^ %Value)

- LvStatus **SetString** (LvDeviceFtr Feature, String^ Value)
- LvStatus **GetBuffer** (LvDeviceFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvDeviceFtr Feature, SizeT% Size)
- LvStatus **SetBuffer** (LvDeviceFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvDeviceFtr Feature, IntPtr% pValue)
- LvStatus **SetPtr** (LvDeviceFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvDeviceFtr Feature, LvEnum% Value)
- LvStatus **SetEnum** (LvDeviceFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvDeviceFtr Feature, String^% SymbolicName)
- LvStatus **SetEnumStr** (LvDeviceFtr Feature, String^ SymbolicName)
- LvStatus **GetEnumValByStr** (LvDeviceFtr Feature, String^ SymbolicName, LvEnum% Value) new
- LvStatus **GetEnumValByStr** (LvDeviceFtr Feature, String^ SymbolicName, LvEnum% Value, LvFtr↩Access% FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvDeviceFtr Feature, LvEnum Value, String^% SymbolicName) new
- LvStatus **GetEnumStrByVal** (LvDeviceFtr Feature, LvEnum Value, String^% SymbolicName, LvFtr↩Access% FtrAccess) new
- LvStatus **CmdExecute** (LvDeviceFtr Feature) new
- LvStatus **CmdExecute** (LvDeviceFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvDeviceFtr Feature, Boolean% IsDone)
- LvStatus **GetAccess** (LvDeviceFtr Feature, LvFtrAccess% FtrAccess)
- LvStatus **GetVisibility** (LvDeviceFtr Feature, LvFtrVisibility% FtrVisibility)
- LvStatus **GetInfo** (LvDeviceFtr Feature, LvFtrInfo FtrInfo, Int32% Info) new
- LvStatus **GetInfo** (LvDeviceFtr Feature, LvFtrInfo FtrInfo, Int32% Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvDeviceFtr Feature, LvFtrInfo FtrInfo, String^% InfoStr) new
- LvStatus **GetInfoStr** (LvDeviceFtr Feature, LvFtrInfo FtrInfo, String^% InfoStr, Int32 Param) new
- LvStatus RegisterFeatureCallback (LvDeviceFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩FeatureParam)

Static Public Member Functions

- static LvStatus Open (LvInterface^ Interface, String^ DeviceId, LvDevice^% Device)
- static LvStatus Open (LvInterface^ Interface, String^ DeviceId, LvDevice^% Device, LvDeviceAccess Device↩Access)
- static LvStatus Close (LvDevice^% Device)

8.3.1 Detailed Description

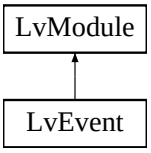
The LvDevice class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/↩Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.3.2 Member Function Documentation

8.3.2.1 LvStatus RegisterFeatureCallback (LvDeviceFtr *Feature,* Boolean *Register,* IntPtr *pUserParam,* IntPtr *pFeatureParam*)

8.4 LvEvent Class Reference

Inheritance diagram for LvEvent:



Public Member Functions

- LvStatus Kill ()
- LvStatus Flush ()
- LvStatus WaitAndGetData (IntPtr pBuffer, SizeT%Size, UInt32 Timeout)
- LvStatus WaitAndGetData (IntPtr pBuffer, SizeT%Size)
- LvStatus WaitAndGetNewBuffer (LvBuffer^%Buffer, UInt32 Timeout)
- LvStatus WaitAndGetNewBuffer (LvBuffer^%Buffer)
- LvStatus GetDataInfo (IntPtr pInBuffer, SizeT InSize, LvEventDataInfo EventDataInfo, IntPtr pBuffer, SizeT%Size, LvInfoDataType%InfoDataType, Int32 Param)
- LvStatus PutData (IntPtr pBuffer, SizeT Size)
- LvStatus SetCallback (Boolean Set, IntPtr pUserParam)
- LvStatus SetCallbackNewBuffer (Boolean Set, IntPtr pUserParam)
- LvStatus StartThread ()
- LvStatus StopThread ()
- LvHEvent GetHandle ()
- LvSystem GetParentSystem ()
- LvDevice GetParentDevice ()
- LvStream GetParentStream ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvEventFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvEventFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^Name, LvEventFtr%Feature)
- Boolean **IsImplemented** (LvEventFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsAvailable** (LvEventFtr Feature)
- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsReadable** (LvEventFtr Feature)
- Boolean **IsWritable** (LvEventFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvEventFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvEventFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvEventFtr Feature, LvFtrType%FtrType) new
- LvStatus **GetType** (LvEventFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvEventFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvEventFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvEventFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvEventFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvEventFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvEventFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvEventFtr Feature, Int64%Value)
- LvStatus **SetInt64** (LvEventFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvEventFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvEventFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvEventFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvEventFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvEventFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvEventFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvEventFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvEventFtr Feature, Double Value)

- LvStatus **GetFloatRange** (LvEventFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvEventFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvEventFtr Feature, String^%Value)
- LvStatus **SetString** (LvEventFtr Feature, String^Value)
- LvStatus **GetBuffer** (LvEventFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvEventFtr Feature, SizeT%Size)
- LvStatus **SetBuffer** (LvEventFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvEventFtr Feature, IntPtr%pValue)
- LvStatus **SetPtr** (LvEventFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvEventFtr Feature, LvEnum%Value)
- LvStatus **SetEnum** (LvEventFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvEventFtr Feature, String^%SymbolicName)
- LvStatus **SetEnumStr** (LvEventFtr Feature, String^SymbolicName)
- LvStatus **GetEnumValByStr** (LvEventFtr Feature, String^SymbolicName, LvEnum%Value) new
- LvStatus **GetEnumValByStr** (LvEventFtr Feature, String^SymbolicName, LvEnum%Value, LvFtr↩Access%FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvEventFtr Feature, LvEnum Value, String^%SymbolicName) new
- LvStatus **GetEnumStrByVal** (LvEventFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩Access%FtrAccess) new
- LvStatus **CmdExecute** (LvEventFtr Feature) new
- LvStatus **CmdExecute** (LvEventFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvEventFtr Feature, Boolean%IsDone)
- LvStatus **GetAccess** (LvEventFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus **GetVisibility** (LvEventFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus **GetInfo** (LvEventFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus **GetInfo** (LvEventFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvEventFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus **GetInfoStr** (LvEventFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus **RegisterFeatureCallback** (LvEventFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩FeatureParam)

Static Public Member Functions

- static LvStatus Open (LvSystem^System, LvEventType EventType, LvEvent^%Event)
- static LvStatus Open (LvDevice^Device, LvEventType EventType, LvEvent^%Event)
- static LvStatus Open (LvStream^Stream, LvEventType EventType, LvEvent^%Event)
- static LvStatus Close (LvEvent^%Event)

Public Attributes

- event LvEventHandler OnEvent
- event LvEventNewBufferHandler OnEventNewBuffer

8.4.1 Detailed Description

The LvEvent class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/↩Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.4.2 Member Data Documentation

8.4.2.1 event LvEventHandler OnEvent

The event handler, which is called whenever a new event appears. The handler is equivalent to a callback function in the DLL version.

Important:

SynView calls the event functions from a separate thread, i.e. the function can be called asynchronously, while your application is executing its code somewhere else. Be aware of this.

8.4.2.2 event LvEventNewBufferHandler OnEventNewBuffer

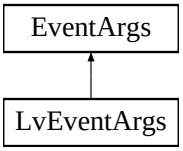
The event handler, which is called whenever a new buffer appears in the output buffer queue. The handler is equivalent to a callback function in the DLL version.

Important:

SynView calls the event functions from a separate thread, i.e. the function can be called asynchronously, while your application is executing its code somewhere else. Be aware of this.

8.5 LvEventArgs Class Reference

Inheritance diagram for LvEventArgs:



Public Attributes

- IntPtr pBuffer
- SizeT Size
- IntPtr pUserParam

8.5.1 Detailed Description

Class for arguments passed to the LvEventHandler() in the LvEvent::OnEvent.

8.5.2 Member Data Documentation

8.5.2.1 IntPtr pBuffer

Pointer to buffer, extracted from the output queue.

8.5.2.2 IntPtr pUserParam

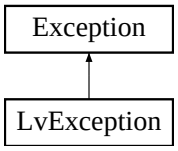
User parameter, supplied in the LvEvent::SetCallback() function.

8.5.2.3 SizeT Size

Buffer size.

8.6 LvException Class Reference

Inheritance diagram for LvException:



Public Member Functions

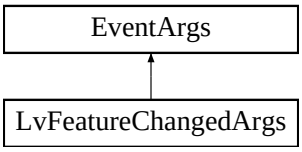
- **LvException** (String^sMessage, LvStatus Number)
- LvStatus **Number** ()

8.6.1 Detailed Description

All exceptions thrown from SynView Net. Class Library are based on this type. Exceptions are used only if the LvLibrary::ThrowErrorEnable property is set to true.

8.7 LvFeatureChangedArgs Class Reference

Inheritance diagram for LvFeatureChangedArgs:



Public Attributes

- IntPtr pUserParam
- IntPtr pFeatureParam
- String Name

8.7.1 Detailed Description

Class for arguments passed to the LvFeatureChangedHandler().

8.7.2 Member Data Documentation

8.7.2.1 String Name

The string ID of the feature.

8.7.2.2 IntPtr pFeatureParam

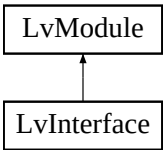
The pFeatureParam passed in the LvSystem::RegisterFeatureCallback(). It is usually used to identify the feature, which has changed.

8.7.2.3 IntPtr pUserParam

The pUserParam, supplied in the LvSystem::RegisterFeatureCallback() function.

8.8 LvInterface Class Reference

Inheritance diagram for LvInterface:



Public Member Functions

- LvStatus UpdateDeviceList ()
- LvStatus UpdateDeviceList (UInt32 Timeout)
- LvStatus GetNumberOfDevices (UInt32%Devices)
- LvStatus GetDeviceId (UInt32 Index, String^%DeviceId)
- LvStatus FindDevice (LvFindBy FindBy, String^FindStr, String^%DeviceId)
- LvHInterface GetHandle ()
- LvStatus OpenDevice (String^DeviceId, LvDevice^%Device)
- LvStatus OpenDevice (String^DeviceId, LvDevice^%Device, LvDeviceAccess DeviceAccess)
- LvStatus CloseDevice (LvDevice^%Device)
- LvSystem GetParentSystem ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvInterfaceFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvInterfaceFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^Name, LvInterfaceFtr%Feature)
- Boolean **IsImplemented** (LvInterfaceFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsAvailable** (LvInterfaceFtr Feature)
- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsReadable** (LvInterfaceFtr Feature)
- Boolean **IsWritable** (LvInterfaceFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvInterfaceFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvInterfaceFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvInterfaceFtr Feature, LvFtrType%FtrType) new
- LvStatus **GetType** (LvInterfaceFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvInterfaceFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvInterfaceFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvInterfaceFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvInterfaceFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvInterfaceFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvInterfaceFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvInterfaceFtr Feature, Int64%Value)

- LvStatus **SetInt64** (LvInterfaceFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvInterfaceFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvInterfaceFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvInterfaceFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvInterfaceFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvInterfaceFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvInterfaceFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvInterfaceFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvInterfaceFtr Feature, Double Value)
- LvStatus **GetFloatRange** (LvInterfaceFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvInterfaceFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvInterfaceFtr Feature, String^%Value)
- LvStatus **SetString** (LvInterfaceFtr Feature, String^Value)
- LvStatus **GetBuffer** (LvInterfaceFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvInterfaceFtr Feature, SizeT%Size)
- LvStatus **SetBuffer** (LvInterfaceFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvInterfaceFtr Feature, IntPtr%pValue)
- LvStatus **SetPtr** (LvInterfaceFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvInterfaceFtr Feature, LvEnum%Value)
- LvStatus **SetEnum** (LvInterfaceFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvInterfaceFtr Feature, String^%SymbolicName)
- LvStatus **SetEnumStr** (LvInterfaceFtr Feature, String^SymbolicName)
- LvStatus **GetEnumValByStr** (LvInterfaceFtr Feature, String^SymbolicName, LvEnum%Value) new
- LvStatus **GetEnumValByStr** (LvInterfaceFtr Feature, String^SymbolicName, LvEnum%Value, LvFtr↩Access%FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvInterfaceFtr Feature, LvEnum Value, String^%SymbolicName) new
- LvStatus **GetEnumStrByVal** (LvInterfaceFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩Access%FtrAccess) new
- LvStatus **CmdExecute** (LvInterfaceFtr Feature) new
- LvStatus **CmdExecute** (LvInterfaceFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvInterfaceFtr Feature, Boolean%IsDone)
- LvStatus **GetAccess** (LvInterfaceFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus **GetVisibility** (LvInterfaceFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus **GetInfo** (LvInterfaceFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus **GetInfo** (LvInterfaceFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvInterfaceFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus **GetInfoStr** (LvInterfaceFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus **RegisterFeatureCallback** (LvInterfaceFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩FeatureParam)

Static Public Member Functions

- static LvStatus Open (LvSystem^System, String^InterfaceId, LvInterface^%Interface)
- static LvStatus Close (LvInterface^%Interface)

8.8.1 Detailed Description

The LvInterface class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/↩Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.9 LvipColorCorrectionMatrix Class Reference

Public Member Functions

- LvStatus SetSaturation (Double Saturation)
- Double **GetMatrixValue** (Int32 i, Int32 j)
- void **SetMatrixValue** (Int32 i, Int32 j, Double Value)

Public Attributes

- internal **__pad0__**: double* m_pdMatrix
- int * **m_piMatrix**

8.10 LvipConvolutionMatrix Class Reference

Public Member Functions

- LvStatus Set3x3Sharpening (Double Factor, Boolean Full)
- Double **GetMatrixValue** (Int32 i, Int32 j)
- void **SetMatrixValue** (Int32 i, Int32 j, Double Value)

Public Attributes

- internal **__pad0__**: int* m_piMatrix
- double * **m_pdMatrix**

8.11 LvipImage Class Reference

Public Member Functions

- LvStatus Initialize (UInt32 Width, UInt32 Height, LvPixelFormat PixelFormat, LvipImgAttr Attributes)
- void **set** (UInt32 Value)
- void **set** (LvipImgAttr Attributes)
- void **set** (IntPtr pData)
- LvStatus AllocateImageData ()
- LvStatus DeallocateImageData ()
- LvStatus CopyFromBmpInfo (IntPtr pBmpInfo)
- LvStatus CopyToBmpInfo (IntPtr pBmpInfo)
- LvStatus FillWithColor (Byte Red, Byte Green, Byte Blue)
- LvStatus FillWithColor (UInt32 ColorRGB)
- LvStatus ApplyLut (LvipImage^DstImage, LvipOption Options, LvipLut^Lut)
- LvStatus ApplyLut (LvipLut^Lut)
- LvStatus CalcWbFactors (UInt32%FactorRed, UInt32%FactorGreen, UInt32%FactorBlue)
- LvStatus CalcWbFactors (UInt32%FactorRed, UInt32%FactorGreen, UInt32%FactorBlue, LvipOption Options)
- LvStatus Deinterlace (LvipImage^DstImage, LvipOption Options)
- LvStatus Rotate90 (LvipImage^DstImage, Boolean ClockWise, LvipOption Options, LvipLut^Lut)
- LvStatus Mirror (LvipImage^DstImage, Boolean TopBottomMirror, Boolean LeftRightMirror, LvipOption Options, LvipLut^Lut)
- LvStatus Rotate90AndMirror (LvipImage^DstImage, Boolean ClockWise, Boolean TopBottomMirror, Boolean LeftRightMirror, LvipOption Options, LvipLut^Lut)

- LvStatus ReverseLines ()
- LvStatus ReverseLines (LvipImage^DstImage, LvipOption Options)
- LvStatus CopyArea (LvipImage^DstImage, Int32 DstXOffset, Int32 DstYOffset, Int32 DstWidth, Int32 Dst↩ Height, LvipOption Options)
- LvStatus SaveToTiff (String^FileName, LvipOption Options)
- LvStatus LoadFromTiff (String^FileName, LvipOption Options)
- LvStatus SaveToBmp (String^FileName, LvipOption Options)
- LvStatus LoadFromBmp (String^FileName, LvipOption Options)
- LvStatus SaveToJpeg (String^FileName, UInt32 QualityFactor)
- LvStatus LoadFromJpeg (String^FileName, LvipOption Options)
- LvStatus ConvertToPixelFormat (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus ApplyRgbColorCorrection (LvipImage^DstImage, LvipColorCorrectionMatrix^Matrix, LvipOption Options, LvipLut^Lut)
- LvStatus ApplyRgbColorCorrection (LvipImage^DstImage, LvipColorCorrectionMatrix^Matrix, LvipOption Options)
- LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^Matrix, LvipLut^Lut)
- LvStatus ApplyRgbColorCorrection (LvipColorCorrectionMatrix^Matrix)
- LvStatus Apply3x3Convolution (LvipImage^DstImage, LvipConvolutionMatrix^Matrix, LvipOption Options, LvipLut^Lut)
- LvStatus Apply3x3Convolution (LvipImage^DstImage, LvipConvolutionMatrix^Matrix, LvipOption Options)
- LvStatus ApplyShadingCorrection (LvipImage^DstImage, LvipImage^BlackRefImage, LvipImage^White↩ RefImage, LvipOption Options, LvipLut^Lut)
- LvStatus ApplyShadingCorrection (LvipImage^DstImage, LvipImage^BlackRefImage, LvipImage^White↩ RefImage, LvipOption Options)
- LvStatus BdShowMosaic (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdGreenToGreyscale (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdNearestNeighbour (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdNearestNeighbour (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdBilinearInterpolation (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdBilinearColorCorrection (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdVariableGradients (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdVariableGradients (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options, LvipLut^Lut)
- LvStatus BdPixelGrouping (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus BdEncodeToBayer (LvipImage^DstImage, LvPixelFormat DstPixelFormat, LvipOption Options)
- LvStatus ConvertToWinBmpCompatibleFormat (LvipImage^ConvImage, LvipOption Options, Boolean%Conversion↩ Needed)
- Bitmap CreateGdiPlusBitmap (LvipImage^ConvImage, LvipOption Options)
- Bitmap CreateUnsafeGdiPlusBitmap (LvipImage^ConvImage, LvipOption Options)

Public Attributes

- property UInt32 Width
- property UInt32 Height
- property LvPixelFormat PixelFormat
- property UInt32 BytesPerPixel
- property UInt32 LinePitch
- property LvipImgAttr Attributes
- property UInt32 ImageDataSize
- property IntPtr Data
- property IntPtr ImgInfo
- internal **__pad0__**: LvipImgInfo* m_pImgInfo
- UInt32 **m_dwReverseLineBufferSize**
- Byte * **m_pReverseLineBuffer**

8.11.1 Member Function Documentation

8.11.1.1 LvStatus ConvertToWinBmpCompatibleFormat (LvipImage^ *ConvImage,* LvipOption *Options,* Boolean% *ConversionNeeded*)

Converts the image to format compatible with Windows Device Independent Bitmap (DIB/BMP), if necessary. If the pixel format is not compatible for display, it uses pConvImgInfo for converting the image - mono images to 8-bit color to 24-bit RGB.
If pConvImgInfo is NULL, and the image has non-compatible format, the image is not converted. If the pConvImgInfo is not NULL, it is used for conversion. It is the user responsibility to deallocate this image when not needed anymore.
Recommended: Create empty LvipImage for ConvImage and in the Options specify the LvipOption::ReallocateDst flag. The function will allocate the appropriate image buffer for the conversion.
Next time this function is called it only checks if the buffer is sufficient, if so, it does not reallocate it. Call LvipDeallocateImageData() to deallocate this helper image at the end of your application.

- Supported pixel formats: 8-bit to 16-bit mono, 15-bit RGB, 16-bit RGB, 24-bit RGB, 32-bit RGB.

Parameters

ConvImage	If conversion to compatible format has to be done, there is a need to give as param to initialized conversion image info structure which will be used for the conversion
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.
ConversionNeeded	If conversion needed ...

Returns

0 in case of failure (use LvipGetLastStatus() for details), 1 in case the image was converted, 2 in case no conversion was needed.

8.11.1.2 Bitmap CreateGdiPlusBitmap (LvipImage^ *ConvImage,* LvipOption *Options*)

Creates a GDI+ bitmap from the image (System::Drawing::Bitmap class instance).

To provide an easy possibility to manipulate with the image in the managed code, two methods are available: LvipImage::CreateGdiPlusBitmap() and LvipImage::CreateUnsafeGdiPlusBitmap(). Both methods return the System::Drawing::Bitmap class instance, which provides methods for accessing bitmap data, and such bitmap can also be simply displayed, for example:

```
PictureBoxLive->Image = MyImage->CreateGdiPlusBitmap(ConvImage,
      LvipOption::ReallocateDst);
```

The LvipImage::CreateGdiPlusBitmap() always creates a new Bitmap object and copies into it the current image. Thus the returned object is independent on the source image and persists even after the LvipImage::Data buffer deallocation or change. The disadvantage is that the image data are always copied, which employs the CPU and uses more memory.

The LvipImage::CreateUnsafeGdiPlusBitmap() is faster; it also creates the a new Bitmap object, but if it is not necessary, it does not copy the bitmap data to it; instead it references the LvipImage::Data buffer, where the image data are already stored. The "unsafe" word expresses the fact that the buffer with the image data is not managed and thus the Bitmap object does not know, when the image buffer is disposed. If it happens, the usage of the Bitmap object may lead to a system exception indicating invalid pointer operation.

So whenever you decide to use the LvipImage::CreateUnsafeGdiPlusBitmap(), be sure you do not use the returned object after the end of the lifetime of the LvipImage::Data buffer of the corresponding LvipImage.

Parameters

ConvImage	Class instance for image pixel format conversion. Used only in case the conversion is needed (not all pixel formats are compatible with the GDI+ bitmap).
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the created GDI+ bitmap.

8.11.1.3 Bitmap CreateUnsafeGdiPlusBitmap (LvipImage^ *ConvImage,* LvipOption *Options*)

Creates a GDI+ bitmap from the image (System::Drawing::Bitmap class instance).

To provide an easy possibility to manipulate with the image in the managed code, two methods are available: LvipImage::CreateGdiPlusBitmap() and LvipImage::CreateUnsafeGdiPlusBitmap(). Both methods return the System::Drawing::Bitmap class instance, which provides methods for accessing bitmap data, and such bitmap can also be simply displayed, for example:

```
PictureBoxLive->Image = MyImage->CreateGdiPlusBitmap(ConvImage,
      LvipOption::ReallocateDst);
```

The LvipImage::CreateGdiPlusBitmap() always creates a new Bitmap object and copies into it the current image. Thus the returned object is independent on the source image and persists even after the LvipImage::Data buffer deallocation or change. The disadvantage is that the image data are always copied, which employs the CPU and uses more memory.

The LvipImage::CreateUnsafeGdiPlusBitmap() is faster; it also creates the a new Bitmap object, but if it is not necessary, it does not copy the bitmap data to it; instead it references the LvipImage::Data buffer, where the image data are already stored. The "unsafe" word expresses the fact that the buffer with the image data is not managed and thus the Bitmap object does not know, when the image buffer is disposed. If it happens, the usage of the Bitmap object may lead to a system exception indicating invalid pointer operation.

So whenever you decide to use the LvipImage::CreateUnsafeGdiPlusBitmap(), be sure you do not use the returned object after the end of the lifetime of the LvipImage::Data buffer of the corresponding LvipImage.

Parameters

ConvImage	Class instance for image pixel format conversion. Used only in case the conversion is needed (not all pixel formats are compatible with the GDI+ bitmap).
Options	Options - OR-ed combination of LvipOption. If the LvipOption::ReallocateDst is used, then also can contain attributes from LvipImgAttr for (re)allocated image.

Returns

Returns the created GDI+ bitmap.

8.12 LvipImgInfo Struct Reference

Public Attributes

- UInt32 StructSize
- UInt32 Width
- UInt32 Height
- UInt32 PixelFormat
- UInt32 Attributes
- UInt32 BytesPerPixel

- UInt32 LinePitch
- IntPtr pData
- IntPtr pDataR
- IntPtr pDataG
- IntPtr pDataB

8.12.1 Detailed Description

Image Info structure. Each image handled by the library must be described by the LvipImgInfo structure.
Although you can set the Image Info members directly, it is highly recommended to use the LvipInitImgInfo() function for the structure initialization.

8.12.2 Member Data Documentation

8.12.2.1 UInt32 Attributes

Image attributes. OR-ed definitions from LvipImgAttr definitions.

8.12.2.2 UInt32 BytesPerPixel

Size of one pixel in bytes.

8.12.2.3 UInt32 Height

Height of the image in pixels.

8.12.2.4 UInt32 LinePitch

Size of one line in bytes.
Example:

```
8-bit mono image: LineIncrement = Width;
24-bit RGB image: LineIncrement = Width * 3;
```

However, when the LvipImgAttr::DWordAligned attribute is used, the line increment must be rounded up to whole double-words, so the calculation would then look like this:

```
8-bit mono image: LineIncrement = (Width+3)/4 * 4;
24-bit RGB image: LineIncrement = ((Width*3)+3)/4 * 4;
```

8.12.2.5 IntPtr pData

Pointer to image data. If color planes are not used, this member points to the data of the image. Use LvipAllocate↩ ImageData() to allocate the buffer for the image. If you set the pointer to an existing image, which is not owned by this LvipImgInfo, use the LvipImgAttr::NotDataOwner attribute.

8.12.2.6 IntPtr pDataB

If color planes are used, this member points to the Blue plane data of the image. Use LvipAllocateImageData() to allocate the buffer for the image. If you set the pointer to an existing image, which is not owned by this LvipImgInfo, use the LvipImgAttr::NotDataOwner attribute.

8.12.2.7 IntPtr pDataG

If color planes are used, this member points to the Green plane data of the image. Use LvipAllocateImageData() to allocate the buffer for the image. If you set the pointer to an existing image, which is not owned by this LvipImgInfo, use the LvipImgAttr::NotDataOwner attribute.

8.12.2.8 IntPtr pDataR

If color planes are used, this member points to the Red plane data of the image. Use LvipAllocateImageData() to allocate the buffer for the image. If you set the pointer to an existing image, which is not owned by this LvipImgInfo, use the LvipImgAttr::NotDataOwner attribute.

8.12.2.9 UInt32 PixelFormat

Pixel format of the image which is saved in this structure. One of the LvPixelFormat
In case of color planes, the pixel format applies to one plane, so use only the MONO formats for the planes. For example for 3x8-bit RGB use the LvPixelFormat::Mono8 format.

8.12.2.10 UInt32 StructSize

Size of image info structure. Should be set to the sizeof(LvipImgInfo). This member may be used in the future versions for the compatibility check.

8.12.2.11 UInt32 Width

Width of the image in pixels.

8.13 LvipLut Class Reference

Public Member Functions

- LvipLut (LvipLutType LutType)
- ~LvipLut ()
- LvStatus ResetLut ()
- LvStatus Set8BitLut (array< Byte >^Red, array< Byte >^Green, array< Byte >^Blue)
- LvStatus Get8BitLut (array< Byte >^Red, array< Byte >^Green, array< Byte >^Blue)
- LvStatus Set10BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Get10BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Set12BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Get12BitLut (array< UInt16 >^Red, array< UInt16 >^Green, array< UInt16 >^Blue)
- LvStatus Set8BitLutValue (LvEnum LutSelector, UInt32 Index, Byte Value)
- LvStatus Get8BitLutValue (LvEnum LutSelector, UInt32 Index, Byte%Value)
- LvStatus Set10BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16 Value)
- LvStatus Get10BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16%Value)
- LvStatus Set12BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16 Value)
- LvStatus Get12BitLutValue (LvEnum LutSelector, UInt32 Index, UInt16%Value)
- LvStatus AddGammaToLut (Double Gamma)
- LvStatus AddWbToLut (Double FactorRed, Double FactorGreen, Double FactorBlue)
- LvStatus AddOffsetAndGainToLut (Double Offset, Double Gain)
- LvStatus AddBrightnessAndContrastToLut (Double Brightness, Double Contrast)

Public Attributes

- internal **__pad0__**: IntPtr m_hLut

8.13.1 Detailed Description

Represents a LUT (Lookup Table) used in the Image Processing Library. This LUT is internally implemented with additional arrays with precalculated values and values with a higher precision, so depending on the LvipLutType, it can consume up to hundreds of kilobytes of memory.

8.14 LvipOverlay Class Reference

Public Member Functions

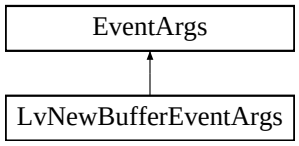
- LvipOverlay (UInt32 Width, UInt32 Height, LvPixelFormat PixelFormat)
- ~LvipOverlay ()
- LvStatus Paint (LvipImage^ DstImage, Int32 XPos, Int32 YPos)
- LvStatus Wipe (UInt32 ColorRGB)
- LvStatus SetTransparentColor (UInt32 ColorRGB)
- UInt32 GetTransparentColor ()
- IntPtr GetDc ()
- LvStatus ReleaseDc ()
- LvStatus SetTextParams (String^ Font, Int32 Size, UInt32 Color, UInt32 OutlineColor, LvipTextAttr Attributes, UInt32 CharSet)
- LvStatus WriteText (String^ Text, Int32 XOffset, Int32 YOffset)
- LvStatus PutBitmap (LvipImage^ Image, Int32 XOffset, Int32 YOffset)
- LvStatus PutBitmapFromBmpFile (String^ FileName, Int32 XOffset, Int32 YOffset)

Public Attributes

- property UInt32 Width
- property UInt32 Height
- property LvPixelFormat PixelFormat
- internal **__pad0__**: IntPtr m_hOverlay

8.15 LvNewBufferEventArgs Class Reference

Inheritance diagram for LvNewBufferEventArgs:



Public Attributes

- LvBuffer Buffer
- IntPtr pUserPointer
- IntPtr pUserParam

8.15.1 Detailed Description

Class for arguments passed to the LvEventNewBufferHandler() in the LvEvent::OnEventNewBuffer.

8.15.2 Member Data Documentation

8.15.2.1 LvBuffer Buffer

LvBuffer class instance representing the buffer extracted from the output buffer queue.

8.15.2.2 IntPtr pUserParam

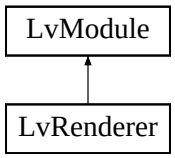
User parameter, supplied in the LvEvent::SetCallback() function.

8.15.2.3 IntPtr pUserPointer

User parameter, supplied in the LvBuffer::Open() function. The same is available in LvBuffer::GetUserPtr()

8.16 LvRenderer Class Reference

Inheritance diagram for LvRenderer:



Public Member Functions

- LvStatus SetWindow (IntPtr hWnd)
- LvStatus CanDisplayImage (LvBuffer^Buffer, UInt32 RenderFlags)
- LvStatus CanDisplayImage (LvBuffer^Buffer)
- LvStatus DisplayImage (LvBuffer^Buffer, UInt32 RenderFlags)
- LvStatus DisplayImage (LvBuffer^Buffer)
- LvStatus Repaint (UInt32 RenderFlags)
- LvStatus Repaint ()
- LvHRenderer GetHandle ()
- LvStream GetParentStream ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvRendererFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvRendererFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^Name, LvRendererFtr%Feature)
- Boolean **IsImplemented** (LvRendererFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsAvailable** (LvRendererFtr Feature)
- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsReadable** (LvRendererFtr Feature)
- Boolean **IsWritable** (LvRendererFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvRendererFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvRendererFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvRendererFtr Feature, LvFtrType%FtrType) new

- LvStatus **GetType** (LvRendererFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvRendererFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvRendererFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvRendererFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvRendererFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvRendererFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvRendererFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvRendererFtr Feature, Int64%Value)
- LvStatus **SetInt64** (LvRendererFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvRendererFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvRendererFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvRendererFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvRendererFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvRendererFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvRendererFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvRendererFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvRendererFtr Feature, Double Value)
- LvStatus **GetFloatRange** (LvRendererFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvRendererFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvRendererFtr Feature, String^%Value)
- LvStatus **SetString** (LvRendererFtr Feature, String^Value)
- LvStatus **GetBuffer** (LvRendererFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvRendererFtr Feature, SizeT%Size)
- LvStatus **SetBuffer** (LvRendererFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvRendererFtr Feature, IntPtr%pValue)
- LvStatus **SetPtr** (LvRendererFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvRendererFtr Feature, LvEnum%Value)
- LvStatus **SetEnum** (LvRendererFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvRendererFtr Feature, String^%SymbolicName)
- LvStatus **SetEnumStr** (LvRendererFtr Feature, String^SymbolicName)
- LvStatus **GetEnumValByStr** (LvRendererFtr Feature, String^SymbolicName, LvEnum%Value) new
- LvStatus **GetEnumValByStr** (LvRendererFtr Feature, String^SymbolicName, LvEnum%Value, LvFtr↩Access%FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvRendererFtr Feature, LvEnum Value, String^%SymbolicName) new
- LvStatus **GetEnumStrByVal** (LvRendererFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩Access%FtrAccess) new
- LvStatus **CmdExecute** (LvRendererFtr Feature) new
- LvStatus **CmdExecute** (LvRendererFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvRendererFtr Feature, Boolean%IsDone)
- LvStatus **GetAccess** (LvRendererFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus **GetVisibility** (LvRendererFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus **GetInfo** (LvRendererFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus **GetInfo** (LvRendererFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvRendererFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus **GetInfoStr** (LvRendererFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus **RegisterFeatureCallback** (LvRendererFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr pFeatureParam)

Static Public Member Functions

- static LvStatus Open (LvStream^Stream, LvRenderer^%Renderer)
- static LvStatus Close (LvRenderer^%Renderer)

8.16.1 Detailed Description

The LvRenderer class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.16.2 Member Function Documentation

8.16.2.1 LvStream GetParentStream ( )

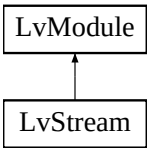
Returns a pointer to the parent class, owning this class instance.

Returns

A pointer to the parent class, owning this class instance.

8.17 LvStream Class Reference

Inheritance diagram for LvStream:



Public Member Functions

- LvStatus GetBufferAt (UInt32 BufferIndex, LvBuffer^%Buffer)
- LvStatus FlushQueue (LvQueueOperation QueueOperation)
- LvStatus Start ()
- LvStatus Start (LvStreamStartFlags StartFlags)
- LvStatus Start (LvStreamStartFlags StartFlags, UInt32 ImagesToAcquire)
- LvStatus Stop ()
- LvStatus Stop (LvStreamStopFlags StopFlags)
- LvHStream GetHandle ()
- LvStatus OpenBuffer (IntPtr pDataPointer, SizeT DataSize, IntPtr pUserPointer, UInt32 Options, LvBuffer^%Buffer)
- LvStatus CloseBuffer (LvBuffer^%Buffer)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^%Event)
- LvStatus CloseEvent (LvEvent^%Event)
- LvStatus OpenRenderer (LvRenderer^%Renderer)
- LvStatus CloseRenderer (LvRenderer^%Renderer)
- LvDevice GetParentDevice ()
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvStreamFtr%Feature) new
- LvStatus **GetFeatureAt** (LvFtrGroup FtrGroup, UInt32 Index, LvStreamFtr%Feature, UInt32%Level) new
- LvStatus **GetFeatureByName** (LvFtrGroup FtrGroup, String^Name, LvStreamFtr%Feature)
- Boolean **IsImplemented** (LvStreamFtr Feature)
- Boolean **IsImplementedByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsAvailable** (LvStreamFtr Feature)

- Boolean **IsAvailableByName** (LvFtrGroup FtrGroup, String^Name)
- Boolean **IsReadable** (LvStreamFtr Feature)
- Boolean **IsWritable** (LvStreamFtr Feature)
- Boolean **IsAvailableEnumEntry** (LvStreamFtr Feature, LvEnum EnumEntry)
- Boolean **IsImplementedEnumEntry** (LvStreamFtr Feature, LvEnum EnumEntry)
- LvStatus **GetType** (LvStreamFtr Feature, LvFtrType%FtrType) new
- LvStatus **GetType** (LvStreamFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus **GetBool** (LvStreamFtr Feature, Boolean%Value)
- LvStatus **SetBool** (LvStreamFtr Feature, Boolean Value)
- LvStatus **GetInt32** (LvStreamFtr Feature, Int32%Value)
- LvStatus **SetInt32** (LvStreamFtr Feature, Int32 Value)
- LvStatus **GetInt32Range** (LvStreamFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus **GetInt32Range** (LvStreamFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus **GetInt64** (LvStreamFtr Feature, Int64%Value)
- LvStatus **SetInt64** (LvStreamFtr Feature, Int64 Value)
- LvStatus **GetInt64Range** (LvStreamFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetInt64Range** (LvStreamFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetInt** (LvStreamFtr Feature, Int64%Value)
- LvStatus **SetInt** (LvStreamFtr Feature, Int64 Value)
- LvStatus **GetIntRange** (LvStreamFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus **GetIntRange** (LvStreamFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus **GetFloat** (LvStreamFtr Feature, Double%Value)
- LvStatus **SetFloat** (LvStreamFtr Feature, Double Value)
- LvStatus **GetFloatRange** (LvStreamFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus **GetFloatRange** (LvStreamFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus **GetString** (LvStreamFtr Feature, String^%Value)
- LvStatus **SetString** (LvStreamFtr Feature, String^Value)
- LvStatus **GetBuffer** (LvStreamFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetBufferSize** (LvStreamFtr Feature, SizeT%Size)
- LvStatus **SetBuffer** (LvStreamFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus **GetPtr** (LvStreamFtr Feature, IntPtr%pValue)
- LvStatus **SetPtr** (LvStreamFtr Feature, IntPtr pValue)
- LvStatus **GetEnum** (LvStreamFtr Feature, LvEnum%Value)
- LvStatus **SetEnum** (LvStreamFtr Feature, LvEnum Value)
- LvStatus **GetEnumStr** (LvStreamFtr Feature, String^%SymbolicName)
- LvStatus **SetEnumStr** (LvStreamFtr Feature, String^SymbolicName)
- LvStatus **GetEnumValByStr** (LvStreamFtr Feature, String^SymbolicName, LvEnum%Value) new
- LvStatus **GetEnumValByStr** (LvStreamFtr Feature, String^SymbolicName, LvEnum%Value, LvFtr↩Access%FtrAccess) new
- LvStatus **GetEnumStrByVal** (LvStreamFtr Feature, LvEnum Value, String^%SymbolicName) new
- LvStatus **GetEnumStrByVal** (LvStreamFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩Access%FtrAccess) new
- LvStatus **CmdExecute** (LvStreamFtr Feature) new
- LvStatus **CmdExecute** (LvStreamFtr Feature, UInt32 Timeout) new
- LvStatus **CmdIsDone** (LvStreamFtr Feature, Boolean%IsDone)
- LvStatus **GetAccess** (LvStreamFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus **GetVisibility** (LvStreamFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus **GetInfo** (LvStreamFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus **GetInfo** (LvStreamFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus **GetInfoStr** (LvStreamFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus **GetInfoStr** (LvStreamFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus **RegisterFeatureCallback** (LvStreamFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩FeatureParam)

Static Public Member Functions

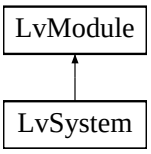
- static LvStatus Open (LvDevice^Device, String^StreamId, LvStream^%Stream)
- static LvStatus Close (LvStream^%Stream)

8.17.1 Detailed Description

The LvStream class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/↩ Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.18 LvSystem Class Reference

Inheritance diagram for LvSystem:



Public Member Functions

- LvStatus UpdateInterfaceList ()
- LvStatus UpdateInterfaceList (UInt32 Timeout)
- LvStatus GetNumberOfInterfaces (UInt32%NumberOfInterfaces)
- LvStatus GetInterfaceId (UInt32 Index, String^%InterfaceId)
- LvStatus FindInterface (LvFindBy FindBy, String^FindStr, String^%InterfaceId)
- LvHSystem GetHandle ()
- LvStatus OpenInterface (String^InterfaceId, LvInterface^%Interface)
- LvStatus CloseInterface (LvInterface^%Interface)
- LvStatus OpenEvent (LvEventType EventType, LvEvent^%Event)
- LvStatus CloseEvent (LvEvent^%Event)
- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvSystemFtr%Feature) new
- LvStatus GetFeatureAt (LvFtrGroup FtrGroup, UInt32 Index, LvSystemFtr%Feature, UInt32%Level) new
- LvStatus GetFeatureByName (LvFtrGroup FtrGroup, String^Name, LvSystemFtr%Feature)
- Boolean IsImplemented (LvSystemFtr Feature)
- Boolean IsImplementedByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsAvailable (LvSystemFtr Feature)
- Boolean IsAvailableByName (LvFtrGroup FtrGroup, String^Name)
- Boolean IsReadable (LvSystemFtr Feature)
- Boolean IsWritable (LvSystemFtr Feature)
- Boolean IsAvailableEnumEntry (LvSystemFtr Feature, LvEnum EnumEntry)
- Boolean IsImplementedEnumEntry (LvSystemFtr Feature, LvEnum EnumEntry)
- LvStatus GetType (LvSystemFtr Feature, LvFtrType%FtrType) new
- LvStatus GetType (LvSystemFtr Feature, LvFtrType%FtrType, LvFtrGui%FtrGui, LvFtrGroup%FtrGroup) new
- LvStatus GetBool (LvSystemFtr Feature, Boolean%Value)
- LvStatus SetBool (LvSystemFtr Feature, Boolean Value)
- LvStatus GetInt32 (LvSystemFtr Feature, Int32%Value)

- LvStatus SetInt32 (LvSystemFtr Feature, Int32 Value)
- LvStatus GetInt32Range (LvSystemFtr Feature, Int32%MinValue, Int32%MaxValue) new
- LvStatus GetInt32Range (LvSystemFtr Feature, Int32%MinValue, Int32%MaxValue, Int32%Increment) new
- LvStatus GetInt64 (LvSystemFtr Feature, Int64%Value)
- LvStatus SetInt64 (LvSystemFtr Feature, Int64 Value)
- LvStatus GetInt64Range (LvSystemFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus GetInt64Range (LvSystemFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus GetInt (LvSystemFtr Feature, Int64%Value)
- LvStatus SetInt (LvSystemFtr Feature, Int64 Value)
- LvStatus GetIntRange (LvSystemFtr Feature, Int64%MinValue, Int64%MaxValue) new
- LvStatus GetIntRange (LvSystemFtr Feature, Int64%MinValue, Int64%MaxValue, Int64%Increment) new
- LvStatus GetFloat (LvSystemFtr Feature, Double%Value)
- LvStatus SetFloat (LvSystemFtr Feature, Double Value)
- LvStatus GetFloatRange (LvSystemFtr Feature, Double%MinValue, Double%MaxValue) new
- LvStatus GetFloatRange (LvSystemFtr Feature, Double%MinValue, Double%MaxValue, Double%Increment) new
- LvStatus GetString (LvSystemFtr Feature, String^%Value)
- LvStatus SetString (LvSystemFtr Feature, String^Value)
- LvStatus GetBuffer (LvSystemFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetBufferSize (LvSystemFtr Feature, SizeT%Size)
- LvStatus SetBuffer (LvSystemFtr Feature, IntPtr pBuffer, SizeT Size)
- LvStatus GetPtr (LvSystemFtr Feature, IntPtr%pValue)
- LvStatus SetPtr (LvSystemFtr Feature, IntPtr pValue)
- LvStatus GetEnum (LvSystemFtr Feature, LvEnum%Value)
- LvStatus SetEnum (LvSystemFtr Feature, LvEnum Value)
- LvStatus GetEnumStr (LvSystemFtr Feature, String^%SymbolicName)
- LvStatus SetEnumStr (LvSystemFtr Feature, String^SymbolicName)
- LvStatus GetEnumValByStr (LvSystemFtr Feature, String^SymbolicName, LvEnum%Value) new
- LvStatus GetEnumValByStr (LvSystemFtr Feature, String^SymbolicName, LvEnum%Value, LvFtr↩Access%FtrAccess) new
- LvStatus GetEnumStrByVal (LvSystemFtr Feature, LvEnum Value, String^%SymbolicName) new
- LvStatus GetEnumStrByVal (LvSystemFtr Feature, LvEnum Value, String^%SymbolicName, LvFtr↩Access%FtrAccess) new
- LvStatus CmdExecute (LvSystemFtr Feature) new
- LvStatus CmdExecute (LvSystemFtr Feature, UInt32 Timeout) new
- LvStatus CmdIsDone (LvSystemFtr Feature, Boolean%IsDone)
- LvStatus GetAccess (LvSystemFtr Feature, LvFtrAccess%FtrAccess)
- LvStatus GetVisibility (LvSystemFtr Feature, LvFtrVisibility%FtrVisibility)
- LvStatus GetInfo (LvSystemFtr Feature, LvFtrInfo FtrInfo, Int32%Info) new
- LvStatus GetInfo (LvSystemFtr Feature, LvFtrInfo FtrInfo, Int32%Info, Int32 Param) new
- LvStatus GetInfoStr (LvSystemFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr) new
- LvStatus GetInfoStr (LvSystemFtr Feature, LvFtrInfo FtrInfo, String^%InfoStr, Int32 Param) new
- LvStatus RegisterFeatureCallback (LvSystemFtr Feature, Boolean Register, IntPtr pUserParam, IntPtr p↩FeatureParam)

Static Public Member Functions

- static LvStatus Open (String^SystemId, LvSystem^%System)
- static LvStatus Close (LvSystem^%System)

8.18.1 Detailed Description

The LvSystem class. @ note For all the SynView module classes you cannot use the new and delete operators directly (the constructor and destructor are private). Instead, the static methods for opening and closing the class instance assure that if the opening is successful, you get a valid pointer, otherwise you get a nullptr(C++)/null(C#)/Nothing(VB) pointer. Also, the closing functions set the pointer back to nullptr(C++)/null(C#)/Nothing(VB). Another advantage is that these functions return a status value, which can clarify the error nature, if the opening or closing fails.

8.18.2 Member Function Documentation

8.18.2.1 **LvStatus CmdExecute (LvSystemFtr *Feature*)** [new]

8.18.2.2 **LvStatus CmdExecute (LvSystemFtr *Feature,* UInt32 *Timeout*)** [new]

8.18.2.3 **LvStatus CmdIsDone (LvSystemFtr *Feature,* Boolean% *IsDone*)**

8.18.2.4 **LvStatus GetAccess (LvSystemFtr *Feature,* LvFtrAccess% *FtrAccess*)**

8.18.2.5 **LvStatus GetBool (LvSystemFtr *Feature,* Boolean% *Value*)**

8.18.2.6 **LvStatus GetBuffer (LvSystemFtr *Feature,* IntPtr *pBuffer,* SizeT *Size*)**

8.18.2.7 **LvStatus GetBufferSize (LvSystemFtr *Feature,* SizeT% *Size*)**

8.18.2.8 **LvStatus GetEnum (LvSystemFtr *Feature,* LvEnum% *Value*)**

8.18.2.9 **LvStatus GetEnumStr (LvSystemFtr *Feature,* String^% *SymbolicName*)**

8.18.2.10 **LvStatus GetEnumStrByVal (LvSystemFtr *Feature,* LvEnum *Value,* String^% *SymbolicName*)** [new]

8.18.2.11 **LvStatus GetEnumStrByVal (LvSystemFtr *Feature,* LvEnum *Value,* String^% *SymbolicName,* LvFtrAccess% *FtrAccess*)** [new]

8.18.2.12 **LvStatus GetEnumValByStr (LvSystemFtr *Feature,* String^ *SymbolicName,* LvEnum% *Value*)** [new]

8.18.2.13 **LvStatus GetEnumValByStr (LvSystemFtr *Feature,* String^ *SymbolicName,* LvEnum% *Value,* LvFtrAccess% *FtrAccess*)** [new]

8.18.2.14 **LvStatus GetFeatureAt (LvFtrGroup *FtrGroup,* UInt32 *Index,* LvSystemFtr% *Feature*)** [new]

8.18.2.15 **LvStatus GetFeatureAt (LvFtrGroup *FtrGroup,* UInt32 *Index,* LvSystemFtr% *Feature,* UInt32% *Level*)** [new]

8.18.2.16 **LvStatus GetFeatureByName (LvFtrGroup *FtrGroup,* String^ *Name,* LvSystemFtr% *Feature*)**

8.18.2.17 **LvStatus GetFloat (LvSystemFtr *Feature,* Double% *Value*)**

8.18.2.18 **LvStatus GetFloatRange (LvSystemFtr *Feature,* Double% *MinValue,* Double% *MaxValue*)** [new]

8.18.2.19 **LvStatus GetFloatRange (LvSystemFtr *Feature,* Double% *MinValue,* Double% *MaxValue,* Double% *Increment*)** [new]

8.18.2.20 **LvStatus GetInfo (LvSystemFtr *Feature,* LvFtrInfo *FtrInfo,* Int32% *Info*)** [new]

8.18.2.21 **LvStatus GetInfo (LvSystemFtr *Feature,* LvFtrInfo *FtrInfo,* Int32% *Info,* Int32 *Param*)** [new]

8.18.2.22 **LvStatus GetInfoStr (LvSystemFtr *Feature,* LvFtrInfo *FtrInfo,* String^% *InfoStr*)** [new]

8.18.2.23 **LvStatus GetInfoStr (LvSystemFtr *Feature,* LvFtrInfo *FtrInfo,* String^% *InfoStr,* Int32 *Param*)** [new]

8.18.2.24 **LvStatus GetInt (LvSystemFtr *Feature,* Int64% *Value*)**

8.18.2.25 **LvStatus GetInt32 (LvSystemFtr *Feature,* Int32% *Value*)**

8.18.2.26 **LvStatus GetInt32Range (LvSystemFtr *Feature,* Int32% *MinValue,* Int32% *MaxValue*)** [new]

8.18.2.27 **LvStatus GetInt32Range (LvSystemFtr *Feature,* Int32% *MinValue,* Int32% *MaxValue,* Int32% *Increment*)** [new]

8.18.2.28 **LvStatus GetInt64 (LvSystemFtr *Feature,* Int64% *Value*)**

8.18.2.29 **LvStatus GetInt64Range (LvSystemFtr *Feature,* Int64% *MinValue,* Int64% *MaxValue*)** [new]

8.18.2.30 **LvStatus GetInt64Range (LvSystemFtr *Feature,* Int64% *MinValue,* Int64% *MaxValue,* Int64% *Increment*)** [new]

8.18.2.31 **LvStatus GetIntRange (LvSystemFtr *Feature,* Int64% *MinValue,* Int64% *MaxValue*)** [new]

8.18.2.32 **LvStatus GetIntRange (LvSystemFtr *Feature,* Int64% *MinValue,* Int64% *MaxValue,* Int64% *Increment*)** [new]

8.18.2.33 **LvStatus GetPtr (LvSystemFtr *Feature,* IntPtr% *pValue*)**

8.18.2.34 **LvStatus GetString (LvSystemFtr *Feature,* String^% *Value*)**

8.18.2.35 **LvStatus GetType (LvSystemFtr *Feature,* LvFtrType% *FtrType*)** [new]

8.18.2.36 **LvStatus GetType (LvSystemFtr *Feature,* LvFtrType% *FtrType,* LvFtrGui% *FtrGui,* LvFtrGroup% *FtrGroup*)** [new]

8.18.2.37 **LvStatus GetVisibility (LvSystemFtr *Feature,* LvFtrVisibility% *FtrVisibility*)**

8.18.2.38 **Boolean IsAvailable (LvSystemFtr *Feature*)**

8.18.2.39 **Boolean IsAvailableByName (LvFtrGroup *FtrGroup,* String^ *Name*)**

8.18.2.40 **Boolean IsAvailableEnumEntry (LvSystemFtr *Feature,* LvEnum *EnumEntry*)**

8.18.2.41 **Boolean IsImplemented (LvSystemFtr *Feature*)**

8.18.2.42 **Boolean IsImplementedByName (LvFtrGroup *FtrGroup,* String^ *Name*)**

8.18.2.43 **Boolean IsImplementedEnumEntry (LvSystemFtr *Feature,* LvEnum *EnumEntry*)**

8.18.2.44 **Boolean IsReadable (LvSystemFtr *Feature*)**

8.18.2.45 **Boolean IsWritable (LvSystemFtr *Feature*)**

8.18.2.46 **LvStatus RegisterFeatureCallback (LvSystemFtr *Feature,* Boolean *Register,* IntPtr *pUserParam,* IntPtr *pFeatureParam*)**

8.18.2.47 **LvStatus** SetBool (**LvSystemFtr** *Feature,* **Boolean** *Value*)

8.18.2.48 **LvStatus** SetBuffer (**LvSystemFtr** *Feature,* IntPtr *pBuffer,* **SizeT** *Size*)

8.18.2.49 **LvStatus** SetEnum (**LvSystemFtr** *Feature,* **LvEnum** *Value*)

8.18.2.50 **LvStatus** SetEnumStr (**LvSystemFtr** *Feature,* **String**^ *SymbolicName*)

8.18.2.51 **LvStatus** SetFloat (**LvSystemFtr** *Feature,* Double *Value*)

8.18.2.52 **LvStatus** SetInt (**LvSystemFtr** *Feature,* **Int64** *Value*)

8.18.2.53 **LvStatus** SetInt32 (**LvSystemFtr** *Feature,* **Int32** *Value*)

8.18.2.54 **LvStatus** SetInt64 (**LvSystemFtr** *Feature,* **Int64** *Value*)

8.18.2.55 **LvStatus** SetPtr (**LvSystemFtr** *Feature,* IntPtr *pValue*)

8.18.2.56 **LvStatus** SetString (**LvSystemFtr** *Feature,* **String**^ *Value*)