- **DS-Ri2/DS-Qi2 SDK Reference**

Contents


SDK Overview
- SDK Execution Environment
- Camera
- Image Information
- Event Notification Method
  - Polling Mode
  - Callback Mode
- Start/End Sequence
- Image Reception Sequence
- Image Format Settings
- Exposure Cancel
- Exposure Signal Output

Installing SDK
- Installation File
  - Redistributable Installation Files (msm)
  - Installation Files for Application Developers
- Shortcuts To Be Generated
  - Start Menu
- Setup Procedure of Driver
  - Device Driver
- Attachment Application
  - Log Analysis Tool
  - Sample Application

SDK External Interface Reference
- Constant
  - Constant Table
  - Error Code Table
  - Other Definition
- Structure
  - Structure Table
  - CAM_Device Structure
  - CAM_FeatureNameRef Structure
  - CAM_Area Structure
  - CAM_Position Structure
  - CAM_TriggerOption Structure
  - CAM_MultiExposureTime Structure
  - CAM_Format Structure
  - CAM_Variant Structure
  - CAM_FeatureValue Structure

- Vector_CAM_FeatureValue Structure
- CAM_FeatureDescElement Structure
- CAM_FeatureDescRange Structure
- CAM_FeatureDescArea Structure
- CAM_FeatureDescPosition Structure
- CAM_FeatureDescTriggerOption Structure
- CAM_FeatureDescFormat Structure
- CAM_FeatureDesc Structure
- CAM_Image Structure
- CAM_ImageInfo Structure
- CAM_ImageInfoEx Structure
- CAM_CMD_GetFrameSize Structure
- CAM_CMD_StartFrameTransfer Structure
- CAM_CMD_IsTransferStarted Structure
- CAM_CMD_FrameDropless Structure
- CAM_CMD_Grouping Structure
- CAM_CMD_GetSdkVersion Structure
- CAM_Command Structure
- CAM_EventImageReceived Structure
- CAM_EventFeatureChanged Structure
- CAM_EventSignal Structure
- CAM_EventTransError Structure
- CAM_EventBusReset Structure
- CAM_Event Structure
- CAM_NoticeTransError Structure
- CAM_NoticeGroup Structure
- CAM_NoticeInfo Structure
- CAM_Notice Structure

Feature

- FeatureTable
- ExposureMode Feature
- ExposureBias Feature
- ExposureTime Feature
- Gain Feature
- MeteringMode Feature
- MeteringArea Feature
- ExposureTimeLimit Feature
- GainLimit Feature
- CaptureMode Feature
- Brightness Feature
- Sharpness Feature
- Hue Feature
- Saturation Feature

- WhiteBalanceRed/WhiteBalanceBlue Feature
- Presets Feature
- TriggerOption Feature
- MultiExposureTime Feature
- SignalExposureEnd Feature
- SignalTriggerReady Feature
- SignalDeviceCapture Feature
- ExposureOutput Feature
- Format Feature
- RoiPosition Feature
- TriggerMode Feature

Method

- Method Table
- CAM_OpenDevices Method
- CAM_CloseDevices Method
- CAM_Open Method
- CAM_Close Method
- CAM_GetAllFeatures Method
- CAM_GetFeatures Method
- CAM_SetFeatures Method
- CAM_GetFeatureDesc Method
- CAM_GetImage Method
- CAM_Command Method
- CAM_EventPolling Method
- CAM_SetEventCallback Method
- CAM_SetNoticeCallback Method

Event

- Event : Image reception event
- Event : Feature change event
- Event : Signal event
- Event : Communication error notification event
- Event : BusReset event
- Notice : Communication error notification event
- Notice : Grouping information notification event
- Notice : Information notification event

# SDK Overview

Thank you for using the Software Development Kit for Nikon DS-Ri2/Qi2 (hereinafter referred to as “SDK”).
This SDK enables you to develop Windows software for DS-Ri2/DS-Qi2, the Nikon Digital Camera System for Microscopy.
This SDK was developed using Microsoft Dynamic Link Library (DLL) technology and is available in a development environment which supports DLL interface.
(The screenshots in this Help document may differ from the actual view designs.)

## SDK Execution Environment

This SDK was developed with Visual Studio 2008 VC++.
It is available on the following environment.

| Item | Description |
|---|---|
| OS | Windows 7 Professional 32-bit and 64-bit |
| Language | English, Japanese |
| Hard Disk | 60 MB or more |
| Memory | 300 MB or more |

## Camera

This SDK can control the following camera types.

- DS-Ri2
- DS-Qi2 (This model specializes in the monochrome image feature of DS-Ri2 and contains cooling function.)

- Each camera supports the following view modes.

◇DS-Ri2

| View Mode | Output Image Size | Image Format |
|---|---|---|
| Full area/Full pixel | 4908x3264 | RGB24, YUV444 |
| ROI1/2 of Full area/Full pixel | 2454x1632 | RGB24, YUV444 |
| Full area/ 1/3 resizing | 1636x1088 | RGB24, YUV444 |
| ROI1/2 of Full area/ 1/3 resizing | 818x544 | RGB24, YUV444 |
| Center Scan/Full pixel | 1608x1608 | RGB24, YUV444 |
| ROI1/2 of Center Scan/Full pixel | 804x804 | RGB24, YUV444 |
| Center Scan/ 1/3 resizing | 536x536 | RGB24, YUV444 |

◇DS-Qi2

| View mode | Output Image Size | Image Format |
|---|---|---|
| Full area/Full pixel | 4908x3264 | Mono16 |
| ROI1/2 of Full area/Full pixel | 2454x1632 | Mono16 |
| Full area/ 1/3 resizing | 1636x1088 | Mono16 |
| ROI1/2 of Full area/ 1/3 resizing | 818x544 | Mono16 |
| Center Scan/Full pixel | 1608x1608 | Mono16 |
| ROI1/2 of Center Scan/Full pixel | 804x804 | Mono16 |
| Center Scan/ 1/3 resizing | 536x536 | Mono16 |

## Image Information

Image Information is appended on the end of image data.

## Event Notification Method

SDK provides two modes, Polling Mode and Callback Mode, for event notification to the application.

## Polling Mode

If application can get any event, this mode invokes CAM_EventPolling method.
There are two modes, <Blocking > and <Non-Blocking>.
Please refer to CAM_EventPolling method for detailed information.

## Callback Mode

If application invokes CAM_SetEventCallback method and then sets callback function for event reception, SDK notices event to the application with the callback function.

## Start/End Sequence

Please invoke methods according to the following procedure to start the application.

- Invoke CAM_OpenDevices method. (Find connected devices.)
- Invoke CAM_Open method. (Open selected camera.)
- Invoke CAM_GetAllFeatures method. (Get Features supported by camera.)
- Invoke CAM_GetFeatureDesc method. (Get attribute values per Feature.)
- If mode is Callback, invoke CAM_SetEventCallback method.
  If mode is Polling, create thread and invoke CAM_EventPolling method.

Please invoke methods according to the following procedure to end the application.

- If mode is Polling, stop it.
- Invoke CAM_Close method. (Close selected camera.)
- Invoke CAM_CloseDevices method. (Release connected devices.)

## Image Reception Sequence

Please invoke methods according to the following procedure to start the application.

<Live>

- Invoke CAM_SetFeatures method. (Set each Feature including Image Format.)
- Invoke CAM_Command method with CAM_CMD_GET_FRAMESIZE. (Get frame size.)
- Invoke CAM_Command method with CAM_CMD_START_FRAMETRANSFER. (Start Image Transfer.)
- (…Image Reception Event)
- Allocate image memory. (It can be allocated beforehand after step .)
- Invoke CAM_GetImage method. (Get Image.)
- Repeat step  - .
- (End Live)
- Invoke CAM_Command method with CAM_CMD_STOP_FRAMETRANSFER. (Stop Image Transfer.)

<Soft Trigger (Capture)>

- Invoke CAM_SetFeatures method. (Set Soft Trigger and each Feature including Image Format.)
- Invoke CAM_Command method with CAM_CMD_GET_FRAMESIZE. (Get frame size.)
- Invoke CAM_Command method with CAM_CMD_START_FRAMETRANSFER. (Start Image Transfer.)
- Invoke CAM_Command method with CAM_CMD_ONEPUSH_SOFTTRIGGER. (Capture)
- (…Image Reception Event)
- Allocate image memory. (It can be allocated beforehand after step .)
- Invoke CAM_GetImage method. (Get Image.)
- Repeat step  - .
- (End Soft Trigger)
- Invoke CAM_Command method with CAM_CMD_STOP_FRAMETRANSFER. (Stop Image Transfer.)

<Hard Trigger>

- Invoke CAM_SetFeatures method. (Set Hard Trigger and each Feature including Image Format.)
- Invoke CAM_Command method with CAM_CMD_GET_FRAMESIZE. (Get frame size.)
- Invoke CAM_Command method with CAM_CMD_START_FRAMETRANSFER. (Start Image Transfer.)

- (Input Hard Trigger.)
- (…Image receive event)
- Allocate image memory. (It can be allocated beforehand after step .)
- Invoke CAM_GetImage method. (Get Image.)
- Repeat step  - .
- (End Hard Trigger.)
- Invoke CAM_Command method with CAM_CMD_STOP_FRAMETRANSFER. (Stop Image Transfer.)

## Image Format Settings

The current metering area, ROI cropping position and Trigger Mode could be out of the range in a new image format if image format is changed and the image size differs from that of the previous one. If they get out of the range, SDK will automatically set default values for the new image format and send FeaturerChanged Event to the application. However, the new range information is not notified from SDK. Therefore, the application needs to get new range information with CAM_GetFeatureDesc method for metering area, ROI cropping position and Trigger Mode after setting image format.

## Exposure Cancel

Depending on the timing of setting Feature and Trigger Cancel command, the camera may cancel the exposure and USB transfer of the current frame during the process. In this case, the camera does not send the last image and cannot correctly inform that to the application. Therefore, please confirm the frame counter of the next image to check if it was gotten after setting Feature or cancel trigger command.

## Exposure Signal Output

There are two types of settings to output ExposureOutput feature, which are ecsoOutput and ecsoLast. As for ecsoOutput, signal is not output if the exposure time is shorter than the following exposure times.

| View Mode | Output Image Size | Exposure Time (msec) |
|---|---|---|
| Full area/Full pixel | 4908x3264 | 64700 |
| ROI1/2 of Full area / Full pixel | 2454x1632 | 32700 |
| Full area/ 1/3 resizing | 1636x1088 | 21800 |
| ROI1/2 of Full area/ 1/3 resizing | 818x544 | 21800 |
| Center Scan/Full pixel | 1608x1608 | 32700 |
| ROI1/2 of Center Scan/Full pixel | 804x804 | 32700 |
| Center Scan/ 1/3 resizing | 536x536 | 21800 |

## Installing SDK

To install this SDK, please run KsCamInstaller64.msi (or setup.exe), the installer package for the Microsoft Windows operating system.
* Hereinafter, this document describes 64-bit edition. However we provide it for 32-bit edition too.
Running this package starts the startup wizard of SDK.
The installer utility is included in Windows 7 as standard utility.
The components of Visual Studio 2008 VC++ runtime libraries, which are needed to run SDK, are also included.

## Installation File

The files to be installed are divided into redistributable installation files and installation files for application developers.

## Redistributable Installation Files (msm)

Please append these to the installer of the application using this SDK.

- **KsCam64.msm**

◇SDK Module (to be installed in the installation folder)

| File | Overview |
|---|---|
| KsCam.dll | Main DLL |

◇Device Driver (to be installed in “Program Files/Nikon/Shared/Drivers/DsCamRev01/” folder)

| File | Overview |
|---|---|
| DsCamRev01.sys | Device Driver (Setup Procedure) |
| DsCamRev01.inf | Device Driver Information |
| DsCamRev01.cat | Device Driver Authorization File |
| WdfCoInstaller01009.dll | |

## Installation Files for Application Developers

Please refer to the VC++ source code included in this for application developers to develop application.

### ■ Files Related to Sample Application

◇Sample Source (to be installed in ”Program Files/Nikon/KsCamSDK/Samples/KsCamExample/” )

| File | Overview |
|---|---|
| KsCamExample.sln etc. | A set of source files |

◇Header (included in Sample Source : Please refer to Sample Source folder/SDK/include/)

| File | Overview |
|---|---|
| KsCam.h | Header file included by application |
| KsCamFeature.h | Feature-related definition |
| KsCamEvent.h | Event-related definition |
| KsCamCommand.h | Command-related definition |
| KsCamImage.h | Image-related definition (including structure of image information) |

◇SDK and libraries (included in Sample Source : Please refer to Sample Source folder/SDK/64bit/)

| File | Overview |
|---|---|
| KsCam.dll | Main DLL |
| KsCam.lib | Library file of Main DLL |

◇Sample Application (to be installed in the installation folder)

| File | Overview |
|---|---|
| KsCamExample.exe | Sample Application |

### ■ Others

◇Redistributable File (to be installed in “Program Files/Nikon/KsCamSDK/Msm/”)

| File | Overview |
|---|---|
| KsCam64.msm | SDK module and driver |

◇Help File (these files : to be installed in “Program Files/Nikon/KsCamSDK/Help/”)

| File | Overview |
|---|---|
| KsCamAPI-J.chm | Japanese help file |
| KsCamAPI-E.chm | English help file |
| HelpChm.ico | Help file icon |

◇Log Analysis Tool (to be installed in “Program Files/Nikon/KsCamSDK/LogViewer/”)

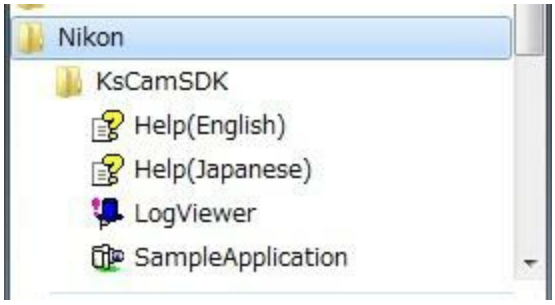
| File | Overview |
|---|---|
| LogViewer.exe | Log Analysis Tool |
| LogViewer.ini | Configuration file for log analysis tool |

## Shortcuts To Be Generated

The installation of this SDK creates shortcuts in the Start menu.

## Start Menu

Setup wizard adds Nikon | KsCamSDK group to the program list of the Start menu.



## Setup Procedure of Driver

Please set up Device Driver.

## Device Driver

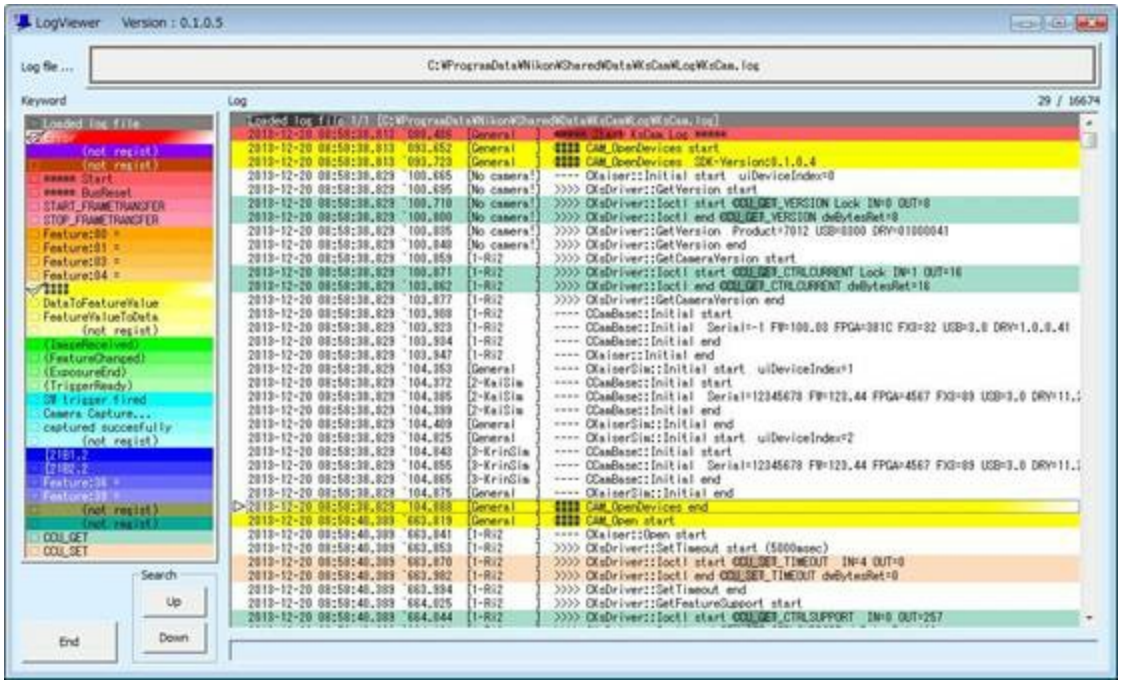
- Setup is completed if the Device Manager is configured as follows.

## Attachment Application

The installation file is provided with Log Analysis Tool and Sample Application using SDK.

## Log Analysis Tool

The file name of the log analysis tool is LogViewer.exe.
The log analysis tool is used to view and search the log file output by SDK and it is useful to analyze the processing procedure inside SDK and the cause of error occurrence.




The contents of the Log list is shown after choosing file (multiply selectable) with the button on the top of the main screen. The left pane shows the Keyword list and it can be searched from the Log list with Search Up/Down buttons below. Right-clicking shows the menu below.

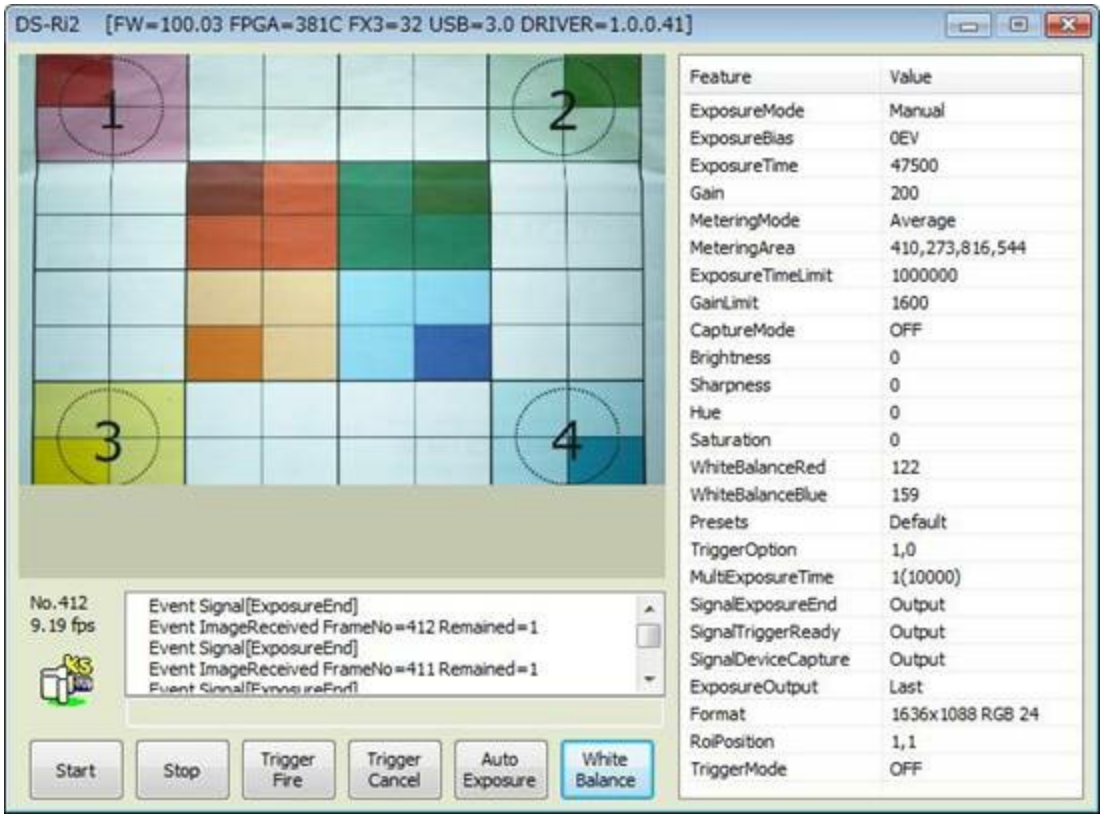
| Name | Overview of Function |
|---|---|
| Sort | Sorts the character string of Log list. |
| Blind | Hides the defined number of the left side of the Log list. |
| Ruler | Shows the thumb of the ruler which is a vertical line of the Log list at the bottom of the Log list. |
| Filter | Please select the filter conditions of the Log list from the items below.<br>Keyword : Only selected keyword is shown.<br>Anti keyword : Selected keyword is excluded.<br>Time… : Only selected range of date and time is shown after selecting the range in the relevant dialog.<br>Please choose one from the items below.<br>Reload : Reloads the selected file. |

| | |
|---|---|
| Log file | Save : Saves the contents of the Log list.<br>Cut : Divides a large-sized file into several files. |
| Settings - Set range | Defines the range of the number of characters for Sort and Blind. |
| Settings - Log list font | Please choose the font for the Log list from the options below.<br>Small<br>Middle<br>Large |
| Settings - Disp lines | Choose the number of lines displayable in the Log list from the options below.<br>All : All the lines are displayed.<br>1000 : Up to 1000 lines are displayed.<br>5000 : Up to 5000 lines are displayed.<br>10000 : Up to 10000 lines are displayed.<br>50000 : Up to 50000 lines are displayed. |
| Settings - Load with sort | Please choose the option for sorting the Log when the Log is loaded.<br>OFF : Log is not sorted.<br>Confirm : The confirmation message is shown to confirm if Log should be sorted or not.<br>Auto : Log is automatically sorted. |
| Settings - Convert time format | Choose the option for converting the character string of date and time into a common format.<br>Auto : Automatically converts it when the file is loaded.<br>Convert and sort : Converts and sorts it at the same time. |
| Settings - Window size | Please choose the following options for adjusting the window size of the main dialog.<br>Tall : The screen is vertically extended to the both top and bottom edges.<br>Wide : The screen is horizontally extended to the right and left edges. |
| Settings - Front | This configures if it is shown in the forefront or not. |
| Settings - Message auto close | This configures if message box should be automatically closed in a second or not. |
| Settings - Dark color | This configures if darker color arrangement is selected or not. |
| Settings - Color for keyword | This configures if it is drawn only for the part of keyword of the Log list instead of the background color |

| | |
|---|---|
| Settings - Regist keyword… | This is a function to change the registration of keyword list. |
| Settings - Keyword collection… | This is a function to choose the list pattern of keyword. |
| Settings- Default path… | This is a function to set default path for loading file. |
| Settings- Password… | This is a function to set password for the customer service to use. |
| Realtime monitor… | This chooses the mode to keep monitoring the file and to display the updated part in the Log list.<br>The following functions are added to the menu while this mode is enabled, although some of the functions limitedly work.<br>Auto scroll : Each time Log is added, the last line is automatically displayed.<br>Keyword filter lock : Only selected keyword is added to the list.<br>Clear log : All the Log is erased. |
| Steal DbWin… | This is a function to steal the log for DebugView for reference. |
| GetLastError… | This is a function to show the description of error code. |
| Reference… | This is a function to refer to FeatureDescription and error code table etc. |
| End | Ends LogViewer. |

## Sample Application

The file name of the Sample Application is KsCamExample.exe.
The Sample Application uses SDK and can test the functions of the camera. Please refer to this application for the use of SDK since it is a module generated by sample source included in the installation.




The upper left pane of the screen is a field to draw image and the frame number and the frame rate are displayed below the lower left hand corner of the field.

The processing history is displayed on the right side of the field.

The bottom part of the screen has the command buttons for Start/Stop of image transfer, Fire/Cancel of Software Trigger (they are enabled when image transfer is started in Software Trigger mode), Auto Exposure, White Balance.
Feature list is on the right side of the screen and configuration change can be applied to the camera with it.

# SDK External Interface Reference

This chapter describes defined values and external interface of SDK.

## Constant

Constant is a defined value which can handle a constant (numerical) value commonly recognizable with SDK as "Name".

## Constant Table

| Constant Name | Definition Code | Value |
|---|---|---|
| ECamDeviceType (Device Type) | ecdtUnknown | 0 |
| | eRi2 | 1 |
| | eRi2_Simulator | 2 |
| | eQi2 | 3 |
| | eQi2_Simulator | 4 |
| ECamFeatureId (Feature ID) | eUnknown | 0 |
| | eExposureMode | 1 |
| | eExposureBias | 2 |
| | eExposureTime | 3 |
| | eGain | 4 |
| | eMeteringMode | 5 |
| | eMeteringArea | 6 |
| | eExposureTimeLimit | 7 |
| | eGainLimit | 8 |
| | eCaptureMode | 9 |
| | eBrightness | 13 |
| | eSharpness | 14 |
| | eHue | 15 |
| | eSaturation | 16 |
| | eWhiteBalanceRed | 18 |
| | eWhiteBalanceBlue | 19 |
| | ePresets | 26 |
| | eTriggerOption | 33 |
| | eMultiExposureTime | 35 |
| | eSignalExposureEnd | 36 |
| | eSignalTriggerReady | 37 |
| | eSignalDeviceCapture | 38 |
| | eExposureOutput | 39 |
| | eFormat | 80 |
| | eRoiPosition | 81 |
| | eTriggerMode | 82 |
| ECamExposureMode | ecemContinuousAE | 0 |

| | | |
|---|---|---|
| (Exposure Mode) | ecemOnePushAE | 1 |
| | ecemManual | 2 |
| | ecemMultiExposureTime | 3 |
| ECamMeteringMode | ecmmAverage | 1 |
| (Metering Mode) | ecmmPeak | 2 |
| ECamPresetsId | ecpiDefault | 0 |
| (Scene Index) | ecpiIndustry_WaferIc | 16 |
| | ecpiIndustry_Metal | 17 |
| | ecpiIndustry_CircuitBoard | 18 |
| | ecpiIndustry_Fpd | 19 |
| | ecpiBio_BrightField | 32 |
| | ecpiBio_He | 33 |
| | ecpiBio_Ela | 34 |
| | ecpiBioLed_BrightField | 48 |
| | ecpiOther_Asbestos | 64 |
| ECamSignalOutput | ecsoOff | 0 |
| (Signal Output) | ecsoOutput | 1 |
| | ecsoLast | 2 |
| ECamFormatColor | ecfcUnknown | 0 |
| (Color Mode) | ecfcRgb24 | 1 |
| | ecfcYuv444 | 2 |
| | ecfcMono16 | 3 |
| ECamFormatSize | ecfsUnknown | 0 |
| (Image Size) | ecfs4908x3264 | 1 |
| | ecfs2454x1632 | 2 |
| | ecfs1636x1088 | 3 |
| | ecfs818x544 | 4 |
| | ecfs1608x1608 | 5 |
| | ecfs804x804 | 6 |
| | ecfs536x536 | 7 |
| ECamTriggerMode | ectmOff | 0 |
| (Trigger Mode) | ectmHard | 1 |
| | ectmSoft | 2 |
| | ectmTriggerMax | 3 |

| | | |
|---|---|---|
| ECamVariantRunType (Feature Value Type) | evrt_unknown | 0 |
| | evrt_int32 | 1 |
| | evrt_uint32 | 2 |
| | evrt_int64 | 3 |
| | evrt_uint64 | 4 |
| | evrt_double | 5 |
| | evrt_bool | 6 |
| | evrt_voidptr | 7 |
| | evrt_wstr | 8 |
| | evrt_Area | 9 |
| | evrt_Position | 10 |
| | evrt_TriggerOption | 11 |
| | evrt_MultiExposureTime | 12 |
| | evrt_Format | 13 |
| ECamFeatureDescType (Feature Attribution Type) | edesc_unknown | 0 |
| | edesc_int32List | 1 |
| | edesc_doubleList | 2 |
| | edesc_ElementList | 3 |
| | edesc_Range | 4 |
| | edesc_Area | 5 |
| | edesc_Position | 6 |
| | edesc_TriggerOption | 7 |
| | edesc_FormatList | 8 |
| ECamGroupCaptureMode (Group Capture Mode) | egcmNoGroup | 0x00 |
| | egcmSoftHard | 0x10 |
| | egcmSoftSoft | 0x20 |
| ECamEventType (Event Type) | ecetUnknown | -1 |
| | ecetImageReceived | 0 |
| | ecetFeatureChanged | 1 |
| | ecetExposureEnd | 2 |
| | ecetTriggerReady | 3 |
| | ecetDeviceCapture | 4 |
| | ecetAeStay | 5 |
| | ecetAeRunning | 6 |
| | ecetAeDisable | 7 |
| | ecetTransError | 8 |

| | | |
|---|---|---|
| | ecetBusReset | 9 |
| | ecetEventTypeMax | 10 |
| ECamEventBusResetCode (Bus Reset Code) | ecebrcHappened | 1 |
| | ecebrcRestored | 2 |
| | ecebrcFailed | 3 |
| ECamNoticeType (Notification Type) | ecntUnknown | -1 |
| | ecntTransError | 0 |
| | ecntGroup | 1 |
| | ecntInfo | 2 |
| | ecntNoticeTypeMax | 3 |
| ECamNoticeGroupCode (Grouping Information Code) | ecngcEventInsufficient | 1 |
| | ecngcSetFeatureError | 2 |
| | ecngcSetTransError | 3 |
| | ecngcSoftTriggerError | 4 |
| | ecngcSetImageFormatError | 5 |
| | ecngcGetImageDataError | 6 |
| | ecngcBusReset | 7 |
| ECamNoticeInfoCode (Notification Information Code) | ecnicTemperature (Currently Unused) | 1 |
| | ecnicComment (Currently Unused) | 2 |

## Error Code Table

| Type Name | Error Code Name | Value |
|---|---|---|
| lx_result (Error Return Value) | LX_OK | 0 |
| | LX_ERR_UNEXPECTED | -1 |
| | LX_ERR_NOTIMPL | -2 |
| | LX_ERR_OUTOFMEMORY | -3 |
| | LX_ERR_INVALIDARG | -4 |
| | LX_ERR_NOINTERFACE | -5 |
| | LX_ERR_POINTER | -6 |
| | LX_ERR_HANDLE | -7 |
| | LX_ERR_ABORT | -8 |
| | LX_ERR_FAIL | -9 |
| | LX_ERR_ACCESSDENIED | -10 |

## Other Definition

| Definition Item | Definition Name | Value |
|---|---|---|
| Max Number of Device Management | CAM_DEVICE_MAX | 100 |
| Error Message Max Length | CAM_ERRMSG_MAX | 256 |
| Version Max Length | CAM_VERSION_MAX | 16 |
| Text Max Length | CAM_TEXT_MAX | 256 |
| Name Max Length | CAM_NAME_MAX | 32 |
| Max Number of Feature ID | CAM_FEA_CAPACITY | 25 |
| Max Length of Variant Character String | CAM_FEA_VARIANT_MAX | 256 |
| Comment Max Length | CAM_FEA_COMMENT_MAX | 64 |
| Max Number of Feature Attribute List | CAM_FEA_DESK_LIST_MAX | 256 |
| Max Number of Multi Exposure Time | CAM_FEA_MULTIEXPOSURETIME_MAX | 15 |
| OnePushAE command | CAM_CMD_ONEPUSH_AE | Command String |
| OnePushWhiteBalance command | CAM_CMD_ONEPUSH_WHITEBALANCE | Command String |
| OnePushSoftTrigger command | CAM_CMD_ONEPUSH_SOFTTRIGGER | Command String |
| OnePushTriggerCancel command | CAM_CMD_ONEPUSH_TRIGGERCANCEL | Command String |
| Get Frame Size command | CAM_CMD_GET_FRAMESIZE | Command String |
| Start Frame Transfer command | CAM_CMD_START_FRAMETRANSFER | Command String |
| Stop Frame Transfer command | CAM_CMD_STOP_FRAMETRANSFER | Command String |
| Image Transfer Confirmation command | CAM_CMD_IS_TRANSFER_STARTED | Command String |
| Frame Dropless Setting Command | CAM_CMD_FRAME_DROPLESS | Command String |
| Grouping Setting Command | CAM_CMD_GROUPING | Command String |
| Get SDK Version command | CAM_CMD_GET_SDKVERSION | Command String |

| | | |
|---|---|---|
| Max Number of Command string | CAM_CMD_STRING_MAX | 64 |

# Structure

This chapter describes structure of information commonly recognizable with SDK.

## Structure Table

| Structure Name | Description |
|---|---|
| CAM_Device | Camera information |
| <For Feature> | |
| CAM_FeatureNameRef | Reference table for feature setting name |
| CAM_Area | Feature setting of area |
| CAM_Position | Feature setting of position |
| CAM_TriggerOption | Feature setting of Trigger Option |
| CAM_MultiExposureTime | Feature setting of multi-exposure time |
| CAM_Format | Feature setting of image format |
| CAM_Variant | Feature setting of Variant value |
| CAM_FeatureValue | Feature value |
| Vector_CAM_FeatureValue | Feature value array |
| <For Feature Attribute> | |
| CAM_FeatureDescElement | Feature attribute of option |
| CAM_FeatureDescRange | Feature attribute of range |
| CAM_FeatureDescArea | Feature attribute of area |
| CAM_FeatureDescPosition | Feature attribute of position |
| CAM_FeatureDescTriggerOption | Feature attribute of Trigger Option |
| CAM_FeatureDescFormat | Feature attribute of image format |
| CAM_FeatureDesc | Feature attribute setting |
| <For Image> | |
| CAM_Image | Image data |
| CAM_ImageInfo | Image information |
| CAM_ImageInfoEx | Extended image information |
| <For Command> | |

| | |
|---|---|
| CAM_CMD_GetFrameSize | For getting frame size |
| CAM_CMD_StartFrameTransfer | For starting frame transfer |
| CAM_CMD_IsTransferStarted | For confirming image transfer |
| CAM_CMD_Grouping | For grouping |
| CAM_CMD_GetSdkVersion | For getting SDK Version |
| CAM_Command | Command request |

<For Event>

| | |
|---|---|
| CAM_EventImageReceived | Image reception event |
| CAM_EventFeatureChanged | Feature change event |
| CAM_EventSignal | Signal event |
| CAM_EventTransError | Communication error event |
| CAM_EventBusReset | Bus Reset event |
| CAM_Event | Event notification |
| CAM_NoticeTransError | Communication error notification |
| CAM_NoticeGroup | Grouping info notification |
| CAM_NoticeInfo | Info notification (Currently Unused) |
| CAM_Notice | Notification |

## CAM_Device Structure

This is a structure to manage device information.

```
typedef struct CAM_Device
{
        ECamDeviceType  eCamDeviceType;
        lx_uint32       uiSerialNo;
        lx_wchar        wszFwVersion[CAM_VERSION_MAX];
        lx_wchar        wszFpgaVersion[CAM_VERSION_MAX];
        lx_wchar        wszFx3Version[CAM_VERSION_MAX];
        lx_wchar        wszUsbVersion[CAM_VERSION_MAX];
        lx_wchar        wszDriverVersion[CAM_VERSION_MAX];
        lx_wchar        wszCameraName[CAM_NAME_MAX];
public:
        CAM_Device() { ZeroMemory(this, sizeof(CAM_Device)); }
} CAM_Device;
```

## CAM_FeatureNameRef Structure

This is a structure to refer to Feature name.

```
typedef struct
{
        EcamFeatureId           eId;
        lx_wchar        wszName[CAM_FEA_COMMENT_MAX];
} CAM_FeatureNameRef;
```

## CAM_Area Structure

This is a structure to set Feature of area.
This is mainly used to set metering area.
This is used if eVarType type of CAM_Variant structure is evrt_Area.

```
struct CAM_Area
{
        lx_uint32 uiLeft;
        lx_uint32 uiTop;
        lx_uint32 uiWidth;
        lx_uint32 uiHeight;
};
```

## CAM_Position Structure

This is a structure to set Feature of position.
This is mainly used to set ROI cropping position.
This is used if eVarType of CAM_Variant structure is evrt_Position.

```
struct CAM_Area
{
        lx_uint32 uiX;
        lx_uint32 uiY;
};
```

## CAM_TriggerOption Structure

This is a structure to set Feature of Trigger Option.
This is used if eVarType of CAM_Variant structure is evrt_TriggerOption.

```
struct CAM_ TriggerOption
{
        lx_uint32 uiFrameCount;
        lx_int32   iDelayTime;
};
```

## CAM_MultiExposureTime Structure

This is a structure to set Feature of multi exposure time.
This is used if eVarType of CAM_Variant structure is evrt_MultiExposureTime.

```
struct CAM_MultiExposureTime
{
        lx_uint32 uiNum;
                                                                lx_uint32
uiExposureTime[CAM_FEA_MULTIEXPOSURETIME_MAX];
};
```

## CAM_Format Structure

This is a structure to set Feature of image format.
This is used if eVarType of CAM_Variant structure is evrt_Format.

```
struct CAM_Format
{
        ECamFormatColor eColor;
        EcamFormatMode eMode;
};
```

## CAM_Variant Structure

This is a structure to set Feature value using Variant. Please refer to a variant in Union via eVarType.

```
struct CAM_Variant
{
        ECamVariantRunType      eVarType;
        union
        {
                lx_int32                i32Value;
                lx_uint32               ui32Value;
                lx_int64                i64Value;
                lx_uint64               ui64Value;
                double                  dValue;
                bool                    bValue;
                void*                   pValue;
                lx_wchar wszValue[CAM_FEA_VARIANT_MAX];
                CAM_Area                stArea;
                CAM_Position            stPosition;
                CAM_TriggerOption       stTriggerOption;
                CAM_MultiExposureTime   stMultiExposureTime;
                CAM_Format              stFormat;
        };
};
```

## CAM_FeatureValue Structure

This is a structure of Feature value.

```
struct CAM_FeatureValue
{
        lx_uint32 uiFeatureId;
        CAM_Variant     stVariant;
         lx_uchar8        ucTransSize;                    // Application don't
use.
};
```

## Vector_CAM_FeatureValue Structure

This is a structure of Feature value array.

```
struct Vector_CAM_FeatureValue
{
        lx_uint32           uiCountUsed;
        lx_uint32           uiCapacity;
        lx_uint32           uiPauseTransfer;
        CAM_FeatureValue*           pstFeatureValue;
};
```

## CAM_FeatureDescElement Structure

This is a structure to set Feature attribute.
This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_ElementList.

```
struct CAM_FeatureDescElement
{
        CAM_Variant         varValue;
        lx_wchar wszComment[CAM_FEA_COMMENT_MAX];
};
```

## CAM_FeatureDescRange Structure

This is a structure to set Feature attribute with range.
This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_Range

```
struct CAM_FeatureDescRange
{
        CAM_Variant     stMin;
        CAM_Variant     stMax;
        CAM_Variant     stRes;
        CAM_Variant     stDef;
};
```

## CAM_FeatureDescArea Structure

This is a structure to set Feature attribute of Area.
This is mainly used to set metering area.
This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_Area.

```
struct CAM_FeatureDescArea
{
        CAM_Area        stMin;
        CAM_Area        stMax;
        CAM_Area        stRes;
        CAM_Area        stDef;
};
```

## CAM_FeatureDescPosition Structure

This is a structure to set Feature attribute of Position.
This is mainly used to set ROI cropping position.
This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_Position.

```
struct CAM_FeatureDescPosition
{
        CAM_Position    stMin;
        CAM_Position    stMax;
        CAM_Position    stRes;
        CAM_Position    stDef;
};
```

## CAM_FeatureDescTriggerOption Structure

This is a structure to set Feature attribute of Trigger Option
This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_TriggerOption.

```
struct CAM_FeatureDescTriggerOption
{
        CAM_FeatureDescRange        stRangeFrameCount;
        CAM_FeatureDescRange        stRangeDelayTime;
};
```

## CAM_FeatureDescFormat Structure

This is a structure to set Feature attribute of image format. This is used if eFeatureDescType of CAM_FeatureDesc structure is edesc_FormatList.

```
struct CAM_FeatureDescFormat
{
        CAM_Format                  stFormat;
        lx_uint32                   uiImageWidth;
        lx_uint32                   uiImageHeight;
        lx_uint32                   uiBitPerPixel;
                                    lx_wchar
wszComment[CAM_FEA_COMMENT_MAX];
        lx_uint32                   uiTriggerListCount;
        CAM_FeatureDescElement      stTriggerList[ectmTriggerMax];
        CAM_FeatureDescArea         stDescArea;
        CAM_FeatureDescPosition     stDescPosition;
};
```

## CAM_FeatureDesc Structure

This is a structure to set Feature attribute using Variant. Please refer to a variant Union via eVarType.

```
struct CAM_FeatureDesc
{
        lx_uint32               uiFeatureId;
        lx_uint32               uiListCount;
        ECamFeatureDescType     eFeatureDescType;
        union
        {
                lx_int32    i32List[CAM_FEA_DESK_LIST_MAX];
                double              dList[CAM_FEA_DESK_LIST_MAX];
                                    CAM_FeatureDescElement
stElementList[CAM_FEA_DESK_LIST_MAX];
                CAM_FeatureDescRange                    stRange;
                CAM_FeatureDescArea                     stArea;
                CAM_FeatureDescPosition         stPosition;
                CAM_FeatureDescTriggerOption    stTriggerOption;
                                    CAM_FeatureDescFormat
stFormatList[CAM_FEA_DESK_LIST_MAX];
        };
};
```

## CAM_Image Structure

This is a structure of image data.
This is parameter of CAM_GetImage method.

```
struct CAM_Image
{
        void*           pDataBuffer;            // include image info
        lx_uint32 uiDataBufferSize;  // set by application.
        lx_uint32 uiImageSize;          // set in SDK (from driver)
        lx_uint32 uiEndTime;            // set in SDK (from driver)
        lx_uint64 uiFrameCount;         // SDK is no care
        lx_uint32 uiRefCount;           // SDK is no care
};
```

## CAM_ImageInfo Structure

This is a structure of image information.

```
typedef struct
{
        //      0
          USHORT usFrameNo;                                   // 0x0000 -> 0xFFFF
            USHORT usTrggerOptionNo;                          // 0=not trigger mode 1 -> 65535
          USHORT usMultiExposureTimeNo;     // 0=not multi exposure time 1 -> 15
        UCHAR   ucReserve000[2];
        //      8
        ULONG   ulExposureTime;   // 100 -> 120000000 (usec)
        UCHAR   ucReserve008[4];
        //      16
        UCHAR   ucReserve016[48];
        //      64
        UCHAR   ucCameraType;               // 0=DS-Ri2, 1=DS-Qi2
        UCHAR   ucImageMode;                // 1 -> 7
            UCHAR    ucImageColor;                  // 1=RGB24, 2=YUV444, 3=Mono16
        UCHAR   ucTriggerMode;     // 0=OFF, 1=Hard, 2=Soft
        ULONG   ulSerialNo;                 //
        //      72
        ULONG   ulFwVersion;                //
        ULONG   ulFpgaVersion;      //
        //      80
        USHORT usFx3Version;                //
        UCHAR   ucReserve080[2];   //
        USHORT usImageWidth;                //
        USHORT usImageHeight;     //
        //      88
        USHORT usRoiLeft;                   //
        USHORT usROITop;                     //
        ULONG   ulImageSize;                //
        //      96
            UCHAR    ucExposureMode; // 0=AE, 1=OnePushAE, 2=Manual, 3=MultiExposureTime
        CHAR    cExposureBias;     // -6 -> 6
        UCHAR   ucTone;             // (...not supported)
```

```
                    UCHAR   ucScene;                    // Scene mode code
(DS-Ri2 only:DS-Qi2=0)
                UCHAR   ucReserve096[4];   //
                //          104
                USHORT usGain;              // 100 -> 6400
                SHORT   sBrightness;                // -50 -> 50
                    CHAR    cSharpness;                 // -3 -> 5 (DS-Ri2
only:DS-Qi2=0)
                UCHAR   ucCaptureMode;    // 0=OFF, 1=ON
                UCHAR   ucAeStay;                   // 0=Running, 1=Stay or
Manual, 2=Disable
                UCHAR   ucMeteringMode;  // 0=Average, 1=Peak
                //          112
                USHORT usMeteringAreaLeft;          //
                USHORT usMeteringAreaTop;           //
                USHORT usMeteringAreaWidth;         //
                USHORT usMeteringAreaHeight; //
                //          120
                  SHORT   sHue;                     // -50 -> 50 (DS-Ri2
only:DS-Qi2=0)
                   SHORT   sSaturation;             // -50 -> 50 (DS-Ri2
only:DS-Qi2=0)
                    USHORT usWhiteBalanceRed;       // 0 -> 799 (DS-Ri2
only:DS-Qi2=0)
                    USHORT usWhiteBalanceBlue;      // 0 -> 799 (DS-Ri2
only:DS-Qi2=0)
                //          128
                USHORT usDefect;                    // (...not supported)
                UCHAR   ucReserve128[6];   //
                //          136
                UCHAR   ucReserve136[120];          //
            } CAM_ImageInfo;

        #define CAM_IMG_INFO_SIZE       sizeof(CAM_ImageInfo)
```

## CAM_ImageInfoEx Structure

This is a structure for extended CAM_ImageInfo structure.
This is enhanced to get each data of image information.

```
struct CAM_ImageInfoEx
{
        union {
                                                lx_uchar8
ucInfo[CAM_IMG_INFO_SIZE];
                CAM_ImageInfo           stInfo;
        };
public:
        void CopyInto(lx_uchar8* pInfo)
        { memcpy(ucInfo, pInfo, CAM_IMG_INFO_SIZE); }
        CAM_ImageInfo* GetInfo(CAM_Image& stImage)
        {
                CopyInto(&((lx_uchar8*)stImage.pDataBuffer)
[stImage.uiImageSize]);
                return &stInfo;
        }
};
```

## CAM_CMD_GetFrameSize Structure

This is a structure to get image size by CAM_Command method.

```
struct CAM_CMD_GetFrameSize
{
        lx_uint32 uiFrameSize;                  // frame size include ImageInfo
        lx_uint32 uiFrameInterval;      // frame interval
        lx_uint32 uiRShutterDelay;    // RSutter delay (usec)
};
```

## CAM_CMD_StartFrameTransfer Structure

This is a structure to start frame transfer by CAM_Command method.

```
struct CAM_CMD_StartFrameTransfer
{
        lx_uint32 uiImageBufferNum;        // 1 - 128 Driver allocate
};
```

## CAM_CMD_IsTransferStarted Structure

This is a structure for the command to confirm image transfer by CAM_Command method.

```
struct CAM_CMD_IsTransferStarted
{
        bool    bStarted;           // true:Started, false:Stopped
};
```

## CAM_CMD_FrameDropless Structure

This is a structure for the command to work out a countermeasure against frame drop by CAM_Command method.

```
struct CAM_CMD_FrameDropless
{
        bool    bSet;                   // true:Set, false:Get
        bool    bOnOff;         // true:ON, false:OFF
};
```

## CAM_CMD_Grouping Structure

This is a structure for the command to set grouping by CAM_Command method.

```
struct CAM_CMD_Grouping
{
        bool            bSet;           // true:Set, false:Get
        lx_uchar8       ucGroup[CAM_DEVICE_MAX];
};
```

## CAM_CMD_GetSdkVersion Structure

This is a structure for the command to get SDK Version by CAM_Command method.

```
struct CAM_CMD_GetSdkVersion
{
        lx_wchar wszSdkVersion[CAM_VERSION_MAX];
};
```

## CAM_Command Structure

This is a structure to serve as parameter of command request by CAM_Command method.

```
struct CAM_Command
{
        lx_wchar wszString[CAM_CMD_STRING_MAX];
        void*           pParameter;
};
```

## CAM_EventImageReceived Structure

This is a structure for image reception event.
This is used if eEventType of CAM_Event structure is ecetImageReceived.

```
struct CAM_EventImageReceived
{
        lx_uint32 uiTick;
        lx_uint32 uiFrameNo;
        lx_uint32 uiRemained;
};
```

## CAM_EventFeatureChanged Structure

This is a structure for Feature change event.
This is used if eEventType of CAM_Event structure is ecetFeatureChanged.

```
struct CAM_EventFeatureChanged
{
        lx_uint32 uiTick;
        lx_uint32 uiFeatureId;
        CAM_Variant     stVariant;
};
```

## CAM_EventSignal Structure

This is a structure for signal event.
This is used if eEventType of CAM_Event structure is ecetExposureEnd／ecetTriggerReady／ecetDeviceCapture／ecetAeStay／ece

```
struct CAM_EventSignal
{
    lx_uint32 uiTick;
    ECamEventType    eEventType;
};
```

## CAM_EventTransError Structure

This is a structure for communication error event.
This is used if eEventType of CAM_Event structure is ecetTransError.

```
struct CAM_EventTransError
{
        lx_uint32 uiTick;
        lx_uint32 uiUsbErrorCode;
    lx_uint32 uiDriverErrorCode;
        lx_uint32 uiReceivedSize;
        lx_uint32 uiSettingSize;
 };
```

## CAM_EventBusReset Structure

This is a structure for Bus Reset event.
This is used if eEventType of CAM_Event structure is ecetBusReset.

```
struct CAM_EventBusReset
{
        lx_uint32                   uiTick;
        ECamEventBusResetCode       eBusResetCode;
        bool                        blmageCleared;
};
```

## CAM_Event Structure

This is a structure for event notification.
Please refer to a variant in Union via eEventType.

```
struct CAM_Event
{
        ECamEventType    eEventType;
        union
        {
                CAM_EventImageReceived  stImageReceived;
                CAM_EventFeatureChanged         stFeatureChanged;
                CAM_EventSignal         stSignal;
                CAM_EventTransError             stTransError;
                CAM_EventBusReset               stBusReset;
        };
};
```

## CAM_NoticeTransError Structure

This is a structure for error notification to device driver.
This is used if eNoticeType of CAM_Notice structure is ecntTransError.

```
struct CAM_NoticeTransError
{
        lx_uint32 uiTick;
        lx_uint32 uiRequestCode;
        lx_uint32 uiCameraErrorCode;
        lx_uint32 uiUsbErrorCode;
        lx_uint32 uiDriverErrorCode;
};
```

## CAM_NoticeGroup Structure

This is a structure for notification of grouping information.
This is used if eNoticeType of CAM_Notice structure is ecntGroup.

```
struct CAM_NoticeGroup
{
        lx_uint32               uiTick;
        ECamNoticeGroupCode     eCode;
        lx_int32                iDetail;
        lx_wchar                wszComment[CAM_TEXT_MAX];
};
```

## CAM_NoticeInfo Structure

This is a structure for notification of information from SDK to the application. This is used if eNoticeType of CAM_Notice structure is ecntInfo.

```
struct CAM_NoticeInfo
{
        lx_uint32           uiTick;
        ECamNoticeInfoCode  eCode;
        lx_int32            iValue;
        double              dValue;
        bool                bValue;
        lx_wchar            wszText[CAM_TEXT_MAX];
};
```

## CAM_Notice Structure

This is a structure for notification.
Please refer to a variant in Union via eNoticeType.

```
struct CAM_Notice
{
        ECamNoticeType   eNoticeType;
        union
        {
                CAM_NoticeTransError            stTransError;
                CAM_NoticeGroup                 stGroup;
                CAM_NoticeInfo          stInfo;
        };
};
```

## Feature

“Feature” is camera setting parameter and the application can set “Feature” such as image format, exposure time and so on. SDK manages the latest values of Features. First, the application gets Feature supported by CAM_GetAllFeatures method with the current setting value. Then it gets attribute values such as setting range and option per Feature by using CAM_GetFeatureDesc method. After that, camera parameter can be set/obtained by using CAM_SetFeatures method and CAM_GetFeatures method. Please get attribute value with CAM_GetFeatureDesc method at all times because setting range may be changed depending on image format as for metering area and ROI cropping position. SDK returns an error if value out of the setting range is set with CAM_SetFeatures method.

## FeatureTable

| Feature | Description |
|---|---|
| ExposureMode | Exposure mode |
| ExposureBias | Exposure compensation value |
| ExposureTime | Exposure time |
| Gain | Gain |
| MeteringMode | Metering mode |
| MeteringArea | Metering area |
| ExposureTimeLimit | Max exposure time during AE |
| GainLimit | Max gain during AE |
| CaptureMode | Capture mode |
| Brightness | Black level |
| Sharpness | Sharpness |
| Hue | Hue |
| Saturation | Saturation |
| WhiteBalanceRed | Red gain of white balance |
| WhiteBalanceBlue | Blue gain of white balance |
| Presets | Scene mode |
| TriggerOption | Trigger Option |
| MultiExposureTime | Multiple exposure time |
| SignalExposureEnd | Exposure end signal |
| SignalTriggerReady | Trigger ready signal |
| SignalDeviceCapture | Device capture signal |
| ExposureOutput | Exposure signal output |
| Format | Image format |
| RoiPosition | ROI cropping position |
| TriggerMode | Trigger mode |

## ExposureMode Feature

- This sets AE mode (exposure control).
- Any of ECamExposureMode can be set.
- The default value is ecemManual.
- If ecemOnePushAE is set while image transfer is stopped, an error is returned.
- Execution time of ecemOnePushAE can vary with exposure time and sample brightness.
- Feature change event is notified to the application if exposure time or gain is change during the processing with both ecemContinuousAE and ecemOnePushAE. If AE convergence state is changed, AE convergence signal event is notified to the application. When the process of ecemOnePushAE ends, ExposureMode is changed to ecemManual and Feature change event is notified to the application.

## ExposureBias Feature

- This sets AE compensation value.
- One of the following values can be set: -1EV, -5/6EV, -2/3EV, -1/2EV, -1/3EV, -1/6EV, 0EV, +1/6EV, +1/3EV, +1/2EV, +2/3EV, +5/6EV and +1EV.
- The default value is 0EV.

## ExposureTime Feature

- This sets exposure time.
- The range of settable values is between 100 usec and 120 sec at 100 usec step intervals.
- The default value is 10 msec.
- Please set the first 3 digits followed by 0. (e.g. 123000 to set 123450)
- It is up to the application to choose the value rounding means: rounding down or off to the $4^{th}$ and subsequent digits.

## Gain Feature

- This sets gain.
- The range of settable values is between 100 and 6400 at 1 step intervals.
- The default value is 100.

## MeteringMode Feature

- This sets metering mode of auto exposure control.
- Any of ECamMeteringMode can be set.
- The default value is ecmmAverage.

## MeteringArea Feature

- This sets Metering Area of Auto Exposure in the format of Left, Top, Width and Height.
- The default values are Left=410, Top=273, Width=816 and Height=544 for DS-Ri2 and Left=403, Top=403, Width=804 and Height=804 for DS-Qi2.
- Please get the setting range with CAM_GetFeatureDesc method due to difference of the setting range depending on image format.
- Please note that the values for Left and Top start from 1 instead of 0.

## ExposureTimeLimit Feature

- This sets the maximum exposure time during auto exposure.
- The range of settable values is between 30 msec and 1 sec at 100 usec step intervals.
- The default value is 1 sec.
- The first 3 digits are valid like ExposureTime Feature.

## GainLimit Feature

- This sets the maximum gain during auto exposure.
- The range of settable values is between 200 and 1600 at 1 step interval.
- The default value is 1600.

## CaptureMode Feature

- This sets whether to optimally tune exposure time and gain after stopping AE when capturing image.
- Either ON or OFF is settable.
- The default value is OFF.
- An error is returned when ON is set if ExposureMode feature is ecemMultiExposureTime.
- An error is returned during ON when ExposureTime feature and Gain feature are set.
- This is mainly used for Capture but is applied to every image if this feature is ON.

## Brightness Feature

- This sets brightness (contrast) of entire image.
- The settable range is between -50 and 50 at 1 step interval.
- The default value is 0.

## Sharpness Feature

- This sets sharpness level of image.
- This is used to get or change the extent of edge enhancement.
- The settable range is between -3 and 5 at 1 step interval..
- The default value is 0.

## Hue Feature

- This sets hue of entire image.
- The settable range is between -50 and 50 at 1 step interval.
- The default value is 0.

## Saturation Feature

- This sets saturation of entire image.
- The settable range is between -50 and 50 at 1 step interval.
- The default value is 0.

## WhiteBalanceRed/WhiteBalanceBlue Feature

- This sets Red/Blue gains of white balance.
- The settable range is between 0 and 799 at 1 step interval.
- The default value is 100 for both Red/Blue gains.
- This Feature is automatically updated if OnePushWhiteBalance command is performed.

## Presets Feature

- This sets scene mode.
- Any of ECamPresetsId can be set.
- The default value is ecpiDefault.

## TriggerOption Feature

- This sets special function of Trigger.
- uiFrameCount (the number of frames to transfer by a single Trigger) and uiDelayTime (the delay time from trigger input until exposure starts) are set to CAM_TriggerOption structure.
- The settable range of uiFrameCount is between 1 and 65535 at 1 step interval.
- The settable range of uiDelayTime is between -16000 usec and 0 usec at 1 step interval.
- The default value is 1 for uiFrameCount and 0 for uiDelayTime.

## MultiExposureTime Feature

- This sets special function of multiple exposure time.
- uiNum (the number of registered exposure time) and uiExposureTime (exposure time) are set to CAM_MultiExposureTime structure.
- The settable range of uiNum is between 1 and 15 at 1 step interval.
- uiExposureTime can be set to the number of uiNum under the setting condition of ExposureTime Feature.
- The default value is 1 for uiNum and 0 for uiShutter.

## SignalExposureEnd Feature

- This sets ON or OFF of exposure end signal event.
- Any of ECamSignalOutput can be set.
- The default value is ecsoOutput.

## SignalTriggerReady Feature

- This sets ON or Off of TriggerReady signal event.
- ecsoOff or ecsoOutput of ECamSignalOutput can be set.
- The default value is ecsoOutput.

## SignalDeviceCapture Feature

- This sets ON or OFF of DeviceCapture signal event.
- ecsoOff or ecsoOutput of ECamSignalOutput can be set.
- The default value is ecsoOutput.

## ExposureOutput Feature

- This sets ON or Off of exposure end signal to external device.
- Any of ECamSignalOutput can be set.
- The default value is ecsoLast.

## Format Feature

- This sets image format.
- The combination of any eColor and any eMode can be set to CAM_Format structure.
- The default value of DS-Ri2 is ecfcRgb24 for eColor and ecfm1636x1088 for eMode.
- The default value of DS-Qi2 is ecfcMono16 for eColor and ecfm1608x1608 for eMode.
- Please set metering area and ROI cropping position as well if Format Feature is set within the limited range. (Feature change event is issued after changing it to the default value by SDK if it is out of the limited range.)

## RoiPosition Feature

- This sets X and Y of ROI cropping position.
- The default value is 1 for both X and Y.
- Please get the setting range with CAM_GetFeatureDesc method due to difference of the setting range depending on image format.

## TriggerMode Feature

- This sets trigger mode.
- Any ECamTriggerMode other than ectmTriggerMax can be set.
- The default value is ectmOff.
- Please get the setting range with CAM_GetFeatureDesc method due to difference of the setting range depending on image format.

## Method

Method is external interface prepared by SDK for the application. There are settings such as camera search at startup, getting and setting the information of Open and Feature, callback function for notification to the application.

## Method Table

| Method Name | Description |
|---|---|
| CAM_OpenDevices | Device Open |
| CAM_CloseDevices | Device Close |
| CAM_Open | Camera Open |
| CAM_Close | Camera Close |
| CAM_GetAllFeatures | Gets all the Feature values from camera. |
| CAM_GetFeatures | Gets Feature value. |
| CAM_SetFeatures | Sets Feature value. |
| CAM_GetFeatureDesc | Gets Feature attribute. |
| CAM_GetImage | Gets image. |
| CAM_Command | Requests command. |
| CAM_EventPolling | Gets event (Polling mode) |
| CAM_SetEventCallback | Sets callback function for event (CallBack mode) |
| CAM_SetNoticeCallback | Sets callback function for notification |

## CAM_OpenDevices Method

Overview : This method finds camera devices connected to PC by opening them.

Argument : OUT lx_uint32& uiDeviceCount Number of camera

OUT CAM_Device** ppstCamDevice Camera Information

Return Value : lx_result

Description : This method finds camera devices connected to PC by opening them.

The number of found camera devices is set to the argument uiDeviceCount and the same number of the argument ppstCamDevice is generated as array to set camera information. Camera simulator is also included in the number of camera.

## CAM_CloseDevices Method

Overview : This method closes devices.
Argument : N/A
Return Value : : lx_result
Description : This method closes devices.

## CAM_Open Method

Overview : This method opens camera.

Argument : : IN const lx_uint32 uiDeviceIndex Camera Information
OUT lx_uint32& uiCameraHandle Camera Handle
IN const lx_uint32 uiErrMsgMaxSize Error Message Text Size
OUT lx_wchar* pwszErrMsg Error Message Text

Return Value : lx_result

Description : This method opens camera.

Index relevant to the camera from the camera information found with CAM_OpenDevices method is set to the argument uiDeviceIndex. If this is normally processed, uiCameraHandle is set to camera handle and returned. Then please invoke the method using this camera handle as argument.

## CAM_Close Method

Overview : This method closes camera.
Argument : IN const lx_uint32 uiCameraHandle Camera Handle
Return Value : lx_result
Description : This method closes camera.

## CAM_GetAllFeatures Method

Overview : This method gets all the Feature information.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

OUT Vector_CAM_FeatureValue& vectFeatureValue All the Feature Value Arrays

Return Value : lx_result

Description : This method gets all the Feature information.

Please invoke this method after setting CAM_FEA_CAPACITY uiCapacity for the argument vectFeatureValue, CAM_FeatureValue structure to pstFeatureValue and pointer with CAM_FEA_CAPACITY of memory allocated. SDK gets the latest setting value from camera and sets it to vectFeatureValue. The application can recognize the Feature set to vectFeatureValue as a supported Feature. This method does not return FeatureValue inside SDK but receives all the FeatureValue again and returns it after setting it inside SDK.

## CAM_GetFeatures Method

Overview : This method gets Feature value.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

INOUT Vector_CAM_FeatureValue& vectFeatureValue Feature Value Array

Return Value : lx_result

Description : This method gets Feature value which is set foruiFeatureId of vectFeatureValue.

SDK does not communicate with camera and sets value managed inside SDK to vectFeatureValue.

## CAM_SetFeatures Method

Overview : This method sets Feature value.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

INOUT Vector_CAM_FeatureValue& vectFeatureValue Feature Value Array

Return Value : : lx_result

Description : : This method sets Feature value which is set for uiFeatureId of vectFeatureValue.

If several Features are set to vectFeatureValue, SDK tunes order of setting and sets them. Please get metering area and ROI cropping position with CAM_GetFeatureDesc method because their range limit may be changed if image format is set. Please get image format and exposure time with the command to get image size if they are set because image size and frame rate may be changed. If one of Format/RoiPosition/TriggerMode is set during image transfer, SDK stops image transfer and then sets the feature. In this case, please restart image transfer by the application if necessary because SDK does not do it. Also, even if is not Format/RoiPosition/TriggerMode, the Feature is set as the status of stopping image transfer temporarily after setting 1 to uiPauseTransfer of vectFeatureValue. In this case, it ensures that requested settings are reflected to the next image.

## CAM_GetFeatureDesc Method

Overview : This method gets Feature attribute.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle
IN lx_uint32 uiFeatureId Feature ID
OUT CAM_FeatureDesc& stFeatureDesc Feature Attribute

Return Value : lx_result

Description : This method gets Feature attribute corresponding to uiFeatureId.

SDK does not communicate with camera and sets values managed inside SDK to stFeatureDesc. Please refer to eFeatureDescType of stFeatureDesc because some Features have their own input ranges and selectable options.

## CAM_GetImage Method

Overview : This method gets image data.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

IN bool bNewRequired TRUE = latest image, FALSE = oldest image

INOUT CAM_Image& stImage Image Data

OUT lx_uint32& uiRemained The Number of Remaining Images

Return Value : lx_result

Description : This method gets image data when image reception event is received.

Please set frame size which is obtained by CAM_CMD_GET_FRAMESIZE command to uiDataBufferSize of stImage and allocate the same size of memory to pDataBuffer before this method is invoked. Please note that Frame size is also changed if Format is changed before getting Frame size. uiRemained does not always match the number of frames remaining in Driver because uiRemained is counted inside SDK.

## CAM_Command Method

Overview : This method requests Command.
Argument : IN const lx_uint32 uiCameraHandle Camera Handle
IN const lx_wchar* wszCommand Command string
INOUT void* pData Data
Return Value : lx_result
Description : SDK allocates processing by the command string specified in wszCommand.
The command is as follows.

<OnePushAE>
Command string is CAM_CMD_ONEPUSH_AE.
It performs OnePushAE.
An error is returned while image is not transferred.
For further information, please refer to the description of ecemOnePushAE of ExposureMode Feature due to the similarity.

<OnePushWhiteBalance>
Command string is CAM_CMD_ONEPUSH_WHITEBALANCE.
It performs OnePushWhiteBalance.
An error is returned while image is not transferred.
I f WhiteBalanceRed/WhiteBalanceBlue Feature is change while processing, Feature change event is notified to the application.

<OnePushSoftTrigger>
Command string is CAM_CMD_ONEPUSH_SOFTTRIGGER.
It performs OnePushSoftTrigger.
Image transfer needs to be started by image transfer start command with TriggerMode Feature set to SoftTrigger before running this command.
An error is returned while image is not transferred.

<OnePushTriggerCancel>
Command string is CAM_CMD_ONEPUSH_TRIGGERCANCEL.
Pressing OnePushSoftTrigger or HardTrigger aborts the running process.
This is especially useful for aborting sequential running with TriggerOption Feature.
An error is returned while TriggerMode is off or image is not transferred.

<Get Frame Size>
Command string is CAM_CMD_GET_FRAMESIZE.
It can get image size (including image information) and frame rate.
Please set CAM_CMD_GetFrameSize structure to parameter.

<Start Frame Transfer>

Command string is CAM_CMD_START_FRAMETRANSFER.

It starts frame transfer.

Please set parameter after specifying the number of frames for the device driver to manage to uiImageBufferNum of CAM_CMD_StartFrameTransfer structure.

Although the settable range of uiImageBufferNum is between 1 and 128, it cannot be set and LX_ERR_OUTOFMEMORY is returned if memory is insufficient. In this case, the settable number is overwritten to uiImageBufferNum and the application tunes the value. Then please retry this command to set the number of frames.

<Stop Frame Transfer>

Command string is CAM_CMD_STOP_FRAMETRANSFER.

It ends frame transfer.

No structure is needed as parameter.

<Image Transfer Confirmation>

Command string is CAM_CMD_ IS_TRANSFER_STARTED.

It gets the status of image being or not being transferred.

Please set CAM_CMD_IsTransferStarted structure as parameter.

Please confirm bStarted of the structure after invoking it.

<Setting to Prevent Frame Drop>

Command string is CAM_CMD_FRAME_DROPLESS.

It sets (or gets) ON/OFF setting for preventing frame drop during Live Live.

Please set CAM_CMD_FrameDropless structure as parameter.

If it is invoked with “false” set to bSet of structure, the setting status of SDK is obtained to bOnOff.

If “true” is set to bSet, please invoke it after setting the status to bOnOff.

If bOnOff is true with long time exposure, SDK automatically changes it to perform Live in MultiExposureTime.

<Grouping Setting>

Command string is CAM_CMD_GROUPING.

It sets (or gets) grouping setting.

Please set CAM_CMD_Grouping structure.

If it is invoked with “false” set to bSet of structure, SDK setting status is obtained to ucGroup.

If “true” is set to bSet, please invoke it after setting the value which is the combination of EcamGroupCaptureMode and group number (0x00～0x0F) with “or” to ucGroup.

An error is returned if it is invoked with any camera in “Open” status.
Cameras with the same ucGroup value are grouped in the same group.
If one of the cameras is opened in the group, the camera is opened as Master and another as Slave.
Slave is not needed to be opened but it can be opened. In the case, request can be output to Slave only although the function is limited.
If requests such as Feature setting and start of image transfer are requested to Master, they are done to Slave too and the cameras are synchronized in a group.
In order to synchronize image transfer too, only SoftTrigger mode is supported, SoftTrigger mode is automatically set by SDK when it is opened and an error is returned to the request to change Trigger mode.
Please invoke it after the application allocates buffer for the number of cameras beforehand because all the image data of the cameras in the group is sequentially stored in the buffer of CAM_GetImage Method if the application gets image per an image reception process.

<Get SDK Version>

Command string is CAM_CMD_ GET_SDKVERSION.
It gets SDK Version.
Please set CAM_CMD_GetSdkVersion structure to parameter.
Please choose 0 for the camera handle of argument due to unnecessity of to specify camera. (This process is performed even if the value other than 0 is set.)

## CAM_EventPolling Method

Overview : This method requests event notification in Polling mode.

Argument : : IN const lx_uint32 uiCameraHandle Camera Handle
IN const HANDLE hStopEvent Polling Stop Event Handle
IN ECamEventType eEventType Event Type
OUT CAM_Event* pstEvent Event Structure

Return Value : lx_result

Description : Please invoke it after setting any ECamEventType to eEventType.
The past event is discarded and overwritten by new event unless the application invokes this method because SDK manages one event per one event type.
Therefore, please process it so as not to stop event by invoking this method again immediately after the application returns from this method.

Polling mode has <Blocking mode> and <Non-Blocking mode>.

<Blocking mode>

If the handle of CreateEvent is handed to hStopEvent, it is called in Blocking mode. In this case, please invoke SetEvent with hStopEvent to abort it because it does not return from this method until the event occurs. The application runs thread and recommends to invoke CAM_EventPolling.

<Non-Blocking mode>

If NULL is set to hStopEvent, it is called in Non-Blocking mode
In this case, it returns after setting an event to pData if there is a waiting event while it returns with an error (LX_ERR_ACCESSDENIED) if there is no waiting event.

## CAM_SetEventCallback Method

Overview : This method sets callback function to request event notification in Callback mode.

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

IN FCAM_EventCallback fCAM_EventCallback Callback Function

IN void* pTransData Transfer Data to the Application

Return Value : lx_result

Description : Please choose Callback function having argument and return value as below.

```
lx_result __cdecl FCAM_EventCallback(
        IN const lx_uint32 uiCameraHandle,
        IN CAM_Event* pstEvent,
        IN void* pTransData)
```

Please allocate process with eEventType inside pstEvent of argument for the application.

pTransData which was set with CAM_SetEventCallback Method is returned for the pTransData of Argument.

## CAM_SetNoticeCallback Method

Overview : This method sets callback function to receive notification from SDK

Argument : IN const lx_uint32 uiCameraHandle Camera Handle

IN FCAM_NoticeCallback fCAM_NoticeCallback Callback Function

IN void* pTransData Transfer Data to the Application

Return Value : lx_result

Description : Please choose Callback function having argument and return value as below.

```
lx_result __cdecl FCAM_NoticeCallback(
        IN const lx_uint32 uiCameraHandle,
        IN CAM_Notice* pstNotice,
        IN void* pTransData)
```

Please allocate process with eNoticeType inside pstNotice of argument for the application.

pTransData which was set with CAM_SetNoticeCallback Method is returned for the pTransData of Argument.

## Event

Event is external interface for notification from SDK to the application.
It is divided into Event and Notice. Event is mainly to notify event from the camera and driver side via SDK while Notice is to notify event occurred inside SDK.

## Event : Image reception event

This event occurs if image is received from camera.
SDK sets ecetImageReceived to eEventType of CAM_Event structure and sends notification to the application after setting information to stImageReceived.
However, uiRemained is a value counted inside SDK and it does not always match the number of images remaining in driver.
The application can get image with CAM_GetImage Method.

## Event : Feature change event

This event occurs if Feature change notification is received from camera or Feature is changed inside SDK.
SDK sets ecetFeatureChanged to eEventType of CAM_Event structure and sends notification to the application after setting information to stFeatureChanged.

## Event : Signal event

This event occurs if various types of signals are received from camera.
SDK sets any value written below to eEventType of CAM_Event structure and sends notification to the application after setting information to stSignal.

| Signal Name | Value |
|---|---|
| Exposure End Signal | ecetExposureEnd |
| Trigger Ready Signal | ecetTriggerReady |
| Device Capture Signal | ecetDeviceCapture |
| AE Convergence Signal | ecetAeStay |
| AE Running Signal | ecetAeRunning |
| Disable AE Convergence Signal | ecetAeDisable |

The application can allocate process with these events as written below (for your information).

Exposure End Signal : This can be used as the timing to close shutter.
Trigger Ready Signal : This can be used as the timing for Trigger input.
Device Capture Signal : It is preferable to run Capture process if possible.
AE Convergence (Running/Disable) : This can be used to judge the AE convergence status.

## Event : Communication error notification event

This event occurs if communication error occurs inside driver.
SDK sets ecetTransError to eEventType of CAM_Event structure and sends notification to the application after setting information to stTransError.

## Event : BusReset event

This event occurs if bus reset occurs inside driver.
SDK sets ecetBusReset to eEventType of CAM_Event structure and sends notification to the application after setting information to stBusReset. Please perform the following process with the application by eBusResetCode of stBusReset.

| eBusResetCode | Process |
|---|---|
| ecebrcHappened | BusReset occurred.<br>The application does not need to cope with anything other than being aware that the subsequent communication will generate an error. |
| ecebrcRestored | Reconnected with camera.<br>Please invoke GetAllFeatures method and synchronize camera with Feature.<br>Images received before bus reset may remain if bImageCleared of stBusReset is FALSE. |
| ecebrcFailed | Could not be reconnected with camera.<br>Please instruct a user that the application should restart camera and be reconnected with the camera. |

## Notice : Communication error notification event

This event occurs if return requested from SDK to driver is an error.
SDK sets ecntTransError to eNoticeType of CAM_Notice structure and sends notification to the application after setting information to stTransError.

## Notice : Grouping information notification event

This event occurs if information to notify or an error to accessing Slave occurs in the process of Master opened in the status of grouping.
Please judge if the request should be aborted or retried when this notification is received because SDK ignores the Slave error.
SDK sets ecntGroup to eNoticeType of CAM_Notice structure and sends notification to the application after setting information to stGroup.

| eCode | iDetail |
|---|---|
| ecngcEventInsufficient | One of the following items is set from ECamEventType.<br>ecetImageReceived<br>ecetExposureEnd<br>ecetTriggerReady |
| ecngcSetFeatureError | FeatureID is set. |
| ecngcSetTransError | Either of the followings is set.<br>0 = Error by Stop transfer<br>1 = Error by Start transfer |
| ecngcSoftTriigerError | Either of the followingsis set.<br>0 = Error by Cancel<br>1 = Error by Fire |
| ecngcSetImageFormatError | 0 is set. |
| ecngcGetImageDataError | The image size (including image information) which was supposed to be obtained is set. |
| ecngcBusReset | One of the ECamEventBusResetCode is set.<br>ecebrcHappened<br>ecebrcRestored<br>ecebrcFailed |

## Notice : Information notification event

This event generates notification item from SDK to the application.
SDK sets ecntInfo to eNoticeType of CAM_Notice structure and sends notification after setting information to stInfo.