April 2023

# API references for Camera Remote SDK

Camera Remote SDK API Reference



SONY

First edition (February 2020)

# Preface

## About this document

The purpose of this document is to list the API specifications for the Camera Remote SDK provided by Sony Corporation.

## Document conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in IETF RFC 2119.

http://www.ietf.org/rfc/rfc2119.txt

For information regarding the latest Camera Remote SDK updates, go to the web site at

http://www.sony.net/CameraRemoteSDK/

# Document history

| Change history | | |
|---|---|---|
| Date | Version | Overview |
| 2020-Feb-06 | 1.00.00 | First version |
| 2020-Jun-18 | 1.00.01 | Just SDK version proceeded with bug fix (no change in the API doc.) |
| 2020-Jul-16 | 1.01.00 | Some of DeviceProperties and Property values are added. |
| 2020-Jul-28 | 1.02.00 | “Supporting products” is updated.<br>Some of DeviceProperties and Property values are added. |
| 2020-Aug-03 | 1.02.00 | “Supporting OS” and “Providing package” are updated. |
| 2020-Sep-15 | 1.02.00 | “Supporting products” is updated. |
| 2020-Oct-15 | 1.02.01 | Just SDK version proceeded with bug fix (no change in the API doc.)<br>Windows version only. |
| 2020-Oct-15 | 1.02.01 | Explanation of Focus_Magnifier_Setting is updated in “CrDeviceProperty” and added in “Tips/Trouble Shooting”. |
| 2020-Dec-08 | 1.03.00 | “Supporting OS” and “Providing package” are updated.<br>Multiple cameras can be controlled by a single SDK.<br>Some of error codes are added. |
| 2021-May-11 | 1.04.00 | “Supporting products” is updated.<br>“Supporting OS” and “Providing package” are updated.<br>Wired LAN connection is added.<br>Some of DeviceProperties and DeviceProperty values are added.<br>Some of error codes are added. |
| 2021-Nov-09 | 1.05.00 | “Supporting products” is updated.<br>Content transfer function via USB connection added.<br>Some of callback functions are added.<br>Some of DeviceProperties and DeviceProperty values are added.<br>Some of error codes are added. |
| 2021-Dec-07 | 1.05.00 | "Function List" is updated for the latest version of ILCE-7RM4A and ILCE-7C. |
| 2022-Oct-12 | 1.06.00 | “Supporting products” is updated.<br>Wired LAN connection by SSH is added.<br>Some of callback functions are added.<br>Some of DeviceProperties and DeviceProperty values are added.<br>Some of error codes are added.<br>“Function List” is updated for the latest version of ILCE-1, ILCE-7SM3 and ILCE-7M4. |
| 2022-Oct-27 | 1.07.00 | “Supporting products” is updated. |
| 2023-Mar-06 | 1.07.00 | “Function List” is updated for ILME-FX3. |
| 2023-Apr-12 | 1.08.00 | “Supporting products” is updated.<br>Some of DeviceProperties and DeviceProperty values are added.<br>Some of error codes are added.<br>Provided as a universal library for macOS.<br>The transfer size of captured still images can now be specified.<br>“Function List” is updated for the latest version of ILCE-7RM5, ILCE-7M4, ILME-FX3 and ILME-FX30. |
| | | |
| | | |

# Contents


About this document ... 3
Document history ... 4
Introduction ... 8
Version, Serial Number, Providing Package ... 8
Version ... 8
Serial number ... 8
Providing Package ... 9
Supporting conditions ... 10
Supporting products and Help Guide URLs ... 10
Supporting physical layer ... 11
Supporting OS ... 12
Environment Setup ... 13
Change the USB Bulk Transfer Rate ... 13
Camera body settings for USB connection ... 16
Install the libusbK driver on Windows ... 16
Camera body settings for wired LAN connection ... 17
Camera body settings for wired LAN connection by SSH ... 19
Uninstallation ... 20
Delete all related folders and files. ... 20
API list ... 21
Function list ... 23
Operational Flow and Sequences ... 36
Initialize and Release Camera Remote SDK ... 37
Enumerate Cameras ... 38
Create a “Camera Object” with information known in advance ... 39
Connect a Camera ... 43
Disconnect a Camera ... 46
Changes in Camera Remote SDK connection status ... 47
Connect/Disconnect multiple cameras ... 48
Get the Camera Properties ... 51
Get the Live View Properties ... 53
Device Properties and Live View Properties ... 54
Change the Camera Properties ... 56
Send a Control Command ... 58
Get a Live View Image ... 59
Capture an Image Sequence ... 61
Change the Store Image Folder and the File Name ... 62
Get the menu display string ... 63
Pull out content stored on media ... 66
Get the MediaProfile ... 70
SDK Properties ... 71
Download and upload setting files ... 72
API Reference ... 73
Initialize ... 74

Release ... 75
CameraObject ... 76
Connection ... 86
Device ... 90
Device Property ... 91
Send Command ... 97
LiveView ... 98
Device Setting ... 105
SDK Version ... 110
SDK Serial Number ... 111
Update SDK Information ... 112
Contents Transfer ... 113
Display string ... 125
Setting file ... 133
MediaProfile ... 137
Lens information ... 140

**Command ... 144**
*CrCommandId ... 144*

**Device Property ... 145**
*CrDeviceProperty ... 145*

**Live View ... 147**
*CrLiveViewProperty ... 147*
*CrFocusFrameInfo ... 149*
*CrMagPosInfo ... 151*
*CrFaceFrameInfo ... 152*
*CrTrackingFrameInfo ... 153*
*CrImageInfo ... 154*
*CrImageDataBlock ... 155*

**Contents Transfer ... 156**
*CrMtpFolderInfo ... 156*
*CrMtpContentsInfo ... 157*

**Display string ... 158**
*CrDisplayStringListInfo ... 158*
*CrDisplayStringType ... 159*

**MediaProfile ... 161**
*CrMediaProfileInfo ... 161*
*CrMediaProfile ... 162*

**Lens Information ... 163**
*CrLensInformation ... 163*
*CrLensInformationType ... 164*

**Callback Interface ... 165**
*IDeviceCallback ... 165*
*ICrCameraObjectInfo ... 167*
*ICrEnumCameraObjectInfo ... 168*

**Status code & Error ... 169**
*Error Category ... 169*
*CrError_None ... 169*
*CrError_Generic ... 170*

*CrError_File* ..... *170*
*CrError_Connect* ..... *171*
*CrError_Memory* ..... *172*
*CrError_Api* ..... *172*
*CrError_Init* ..... *172*
*CrError_Polling* ..... *172*
*CrError_Adaptor* ..... *172*
*CrError_Device* ..... *173*
*CrError_Contents* ..... *173*
*CrWarning* ..... *173*
*CrNotify* ..... *176*

**Parameter description** ..... **177**

**Tips / Trouble Shooting** ..... **282**
Shutter Release ..... 282
Shutter Half Release / Auto Focus ..... 282
Manual Focus ..... 282
Device Property ..... 283
Transfer of shot images preparation ..... 283
Selected Media Format ..... 284
Zoom Operation / Zoom Scale ..... 284
Live View ..... 285
Camera Settings Saving ..... 286
Focus Magnifier Setting ..... 286
About the Monitor DISP(Screen Display) for camera body ..... 288
How to use LensInformation ..... 289
GPS information and shooting image link ..... 290

**More information** ..... **292**
Trademarks and acknowledgements ..... 292

# Introduction

The purpose of this document is to describe the API specifications and information about how to access camera functions and the procedure to establish connection to use the APIs for the Camera Remote SDK.

# Version, Serial Number, Providing Package

## Version

The Camera Remote SDK itself has one version, the app may check this version and change its behavior accordingly.

### Camera Remote SDK version

Camera Remote SDK has its version defined by its specifying functions. The version will be changed if an API is added or deleted. The version also will be changed if a supporting function in any APIs is changed. The Camera Remote SDK version can be obtained by the "GetSDKVersion" API. For details, please see the “GetSDKVersion" API specification.

## Serial number

The Camera Remote SDK itself has a serial number, the app may check this serial number.

### Camera Remote SDK serial number

Camera Remote SDK has its serial number. The Camera Remote SDK serial number can be obtained by the "GetSDKSerial" API. For details, please see the “GetSDKSerial" API specification.

# Providing Package

Camera Remote SDK has following packages.

- Camera Remote SDK for Windows
- Camera Remote SDK for Linux 64bit PC
- Camera Remote SDK for Linux 64bit (ARMv8)
- Camera Remote SDK for Linux 32bit (ARMv7)
- Camera Remote SDK for macOS

# Supporting conditions

Even if the support conditions below are satisfied, it does not guarantee proper operation in all environments.

## Supporting products and Help Guide URLs

Functions and parameters that are not supported by your camera cannot be used even if they are described in the API specification.

Please update each camera to the latest System Software (Firmware) before use.

- ILCE-1 https://helpguide.sony.net/ilc/2040/v1/en/index.html
- ILCE-9M2 https://helpguide.sony.net/ilc/1960/v1/en/index.html
- ILCE-7RM5 https://helpguide.sony.net/ilc/2230/v1/en/index.html
- ILCE-7RM4A https://helpguide.sony.net/ilc/2060/v1/en/index.html
- ILCE-7RM4 https://helpguide.sony.net/ilc/1930/v1/en/index.html
- ILCE-7SM3 https://helpguide.sony.net/ilc/2010/v1/en/index.html
- ILCE-7M4 https://helpguide.sony.net/ilc/2110/v1/en/index.html
- ILCE-7C https://helpguide.sony.net/ilc/2020/v1/en/index.html
- ILME-FX6V/ILME-FX6T (Ver. 3.00 or later)
  https://pro.sony/support/res/manuals/5024/c3bfbc891ee0f149e46d142754fd6aa7/50244581M.pdf
- ILME-FX3 (Ver. 2.00 or later)
  https://helpguide.sony.net/ilc/2210/v1/en/index.html
- ILME-FX30 https://helpguide.sony.net/ilc/2220/v1/en/index.html
- ZV-E1 https://helpguide.sony.net/ilc/2310/v1/en/index.html
- DSC-RX0M2 (Ver. 3.00 or later)
  https://helpguide.sony.net/dsc/1910/v1/en/index.html

Note : In this document, ILME-FX6V/ILME-FX6T will be referred to as ILME-FX6.

# Supporting physical layer

USB, Ethernet(Wired LAN)

| No. | Model Name | USB | | Ethernet (Wired LAN) | |
|---|---|---|---|---|---|
| | | R | C | R | C |
| 1 | ILCE-1 | ✓ | ✓ | ✓ | ✓ |
| 2 | ILCE-9M2 | ✓ | - | ✓ | - |
| 3 | ILCE-7RM5 | ✓ | ✓ | - | - |
| 4 | ILCE-7RM4A | ✓ | ✓ | - | - |
| 5 | ILCE-7RM4 | ✓ | - | - | - |
| 6 | ILCE-7SM3 | ✓ | ✓ | - | - |
| 7 | ILCE-7M4 | ✓ | ✓ | - | - |
| 8 | ILCE-7C | ✓ | ✓ | - | - |
| 9 | ILME-FX6 (Ver. 3.00 or later) | - | - | ✓* | - |
| 10 | ILME-FX3 (Ver. 2.00 or later) | ✓ | ✓ | - | - |
| 11 | ILME-FX30 | ✓ | ✓ | - | - |
| 12 | ZV-E1 | ✓ | ✓ | - | - |
| 13 | DSC-RX0M2 (Ver. 3.00 or later) | ✓ | ✓ | - | - |

“R" refers for RemoteControlMode, “C” refers for ContentsTransferMode,
The ContentsTransferMode feature was added in version 1.05.00.
- See “Connect” for the mode to connect.

*: With SSH authentication. Use a USB Type C wired LAN adaptor. Use of a Gigabit Ethernet compatible adaptor is recommended.

# Supporting OS

- Camera Remote SDK for Windows

  Checked with the environment on “Windows 10 64bit”, “Windows 11 64bit”

- Camera Remote SDK for Linux 64bit PC

  Checked with the environment on “Ubuntu 20.04.1 LTS”, “Ubuntu 22.04.1 LTS”

- Camera Remote SDK for Linux 64bit (ARMv8)

  Checked with the environment below.

| No. | Hardware | CPU | OS |
|---|---|---|---|
| 1 | Jetson Nano Developer Kit B01 | ARMv8 Cortex-A57 | Ubuntu 20.04.1 LTS (GNU/Linux 4.9.140-tegra aarch64) |
| 2 | Raspberry Pi4 Model B (4GB) | ARMv8 Cortex-A72 | Raspberry Pi OS (64 bit) beta test version |

- Camera Remote SDK for Linux 32bit (ARMv7)

  Checked with the environment below.

| No. | Hardware | CPU | OS |
|---|---|---|---|
| 1 | Raspberry Pi2 Model B V1.1 (Broadcom BCM2836) | ARMv7 Cortex-A7 | Raspberry Pi OS (32-bit) with desktop (Version: May 2020) |

Even if the support conditions are satisfied, it does not guarantee proper operation in all environments.

- Camera Remote SDK for macOS

  Checked with the environment on “11.1 or later(Big Sur)” and “12.1 or later(Monterey)” and “13.1 or later(Ventura)”
  Provided as a universal library from version 1.08.00.

# Environment Setup

## Change the USB Bulk Transfer Rate

USB Bulk Transfer Rate should be changed to 150. The way to set it depends on the OS.

This value represents the maximum data size of USB bulk transmission and should be larger than the file size transferred from cameras to the host. (Unit is [MB].)

If you need to adjust memory size adequately, you should set this value to the maximum file size of your camera model.

### Raspberry Pi OS

Open /etc/rc.local with an editor.

Add the command below at the end of the file before "exit 0" to modify Bulk Transfer Rate configuration file.

*Add this command:*

```
sudo sh -c 'echo 150 > /sys/module/usbcore/parameters/usbfs_memory_mb'
```

Save & Close the file and reboot. Make sure that “150” is written in the configuration file.

```
$ cat /sys/module/usbcore/parameters/usbfs_memory_mb
150
```

### Ubuntu (for Embedded)

Open /boot/extlinux/extlinux.conf with an editor.

Change “APPEND ${cbootargs} quiet” to the command below.

*Before:*

```
APPEND ${cbootargs} quiet
```

After:

```
APPEND ${cbootargs} usbcore.usbfs_memory_mb=150 usbcore.autosuspend=-1
```

Save & Close the file and reboot. Make sure that “150” is written in the configuration file.

```
$ cat /sys/module/usbcore/parameters/usbfs_memory_mb
150
```

### Ubuntu (for x86)

Open /etc/default/grub with an editor.

Change “quiet splash” to the command below.

```
Before:
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

```
After:
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash usbcore.usbfs_memory_mb=150"
```

Save & Close the file and update grub.

```
$ sudo update-grub
```

Reboot and make sure that “150” is written in the configuration file.

```
$ cat /sys/module/usbcore/parameters/usbfs_memory_mb
150
```

# Camera body settings for USB connection

When connecting the SDK to the camera via a USB cable, the following settings must be made on the camera itself.

ILCE-1 and DSC-RX0M2 is used as an example here. For other models, refer to "PC Remote Function" in the Help Guide.

**For ILCE-1**

- Please set “Network > Transfer/Remote > PC Remote Function > PC Remote” to “On”.
- The default setting of “Network > Transfer/Remote > PC Remote Function > PC Remote Cnct Method” is “USB”, but if other than “USB” is set, change it to “USB”.

**For DSC-RX0M2**

- Please set “Setup > USB Connection” to “PC Remote”.

# Install the libusbK driver on Windows

If you want to connect via USB on Windows, you need to install the libusbK driver.

Please refer to "0. Preparation-> Installation of libusbK" page of RemoteSampleApp_IM_vx.xx.xx.pdf.

# Camera body settings for wired LAN connection

When connecting the SDK to the camera via a wired LAN, the following settings must be made on the camera itself.

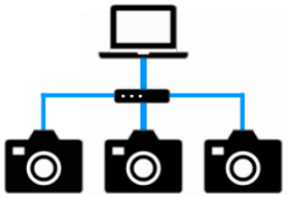
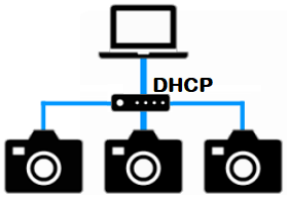
ILCE-1 is used as an example here. For other models, refer to "PC Remote Function" in the Help Guide.

- Please set “Network > Transfer/Remote > PC Remote Function > PC Remote” to “On”.
- The default setting for “Network > Transfer/Remote > PC Remote Function > PC Remote Cnct Method” is “USB”. Please change it to “Wired LAN”.
  After enabling the wired LAN connection, it takes about 30 seconds for the SDK to recognize the camera.
  When connecting via wired LAN, you need to perform the " Network > Transfer/Remote > PC Remote Function > Pairing" operation on the camera to make it memorize the host PC. Once the pairing is established, turn off the camera, pause for about 10 seconds, and then turn it back on again. (The information is stored in the camera when the power is turned off.)
- You can also connect without “Network > Transfer/Remote > PC Remote Function > Pairing”.
  Connections without "Pairing" are possible by setting "Connect without Pairing" to "Enable".
  If you set "Connect without Pairing" to "Enable", unintended third parties may access the camera.
  Sony is not liable for any problems or damage caused by setting "Connect without Pairing" to "Enable".

## IP Address Setting

- The default setting for “Network > Wired LAN > IP Address Setting > IP Address Setting” is “Auto”. If the camera is connecting to a router with a DHCP service, set the setting to "Auto" to automatically assign an IP address. If you want to use a network HUB or connect directly to the host PC, change the setting to "Manual" and set the IP address manually.
- " Network > Wired LAN > IP Address Setting > IP Address Setting > Auto" can also be used when the camera is not connected to a router or similar. In this case, the IP address is determined by the camera itself. The host PC should set its IP address based on the one determined by the camera.

For the combination of connection type and ”IP address setting”, please use the following table to help.

| | Direct | | Use HUB | | Use Router | |
|---|---|---|---|---|---|---|
| |  | |  | |  | |
| | Auto | Manual | Auto | Manual | Auto | Manual |
| Windows | *1 | - | *1 | - | *1 | - |
| macOS | *2 | | | | | |
| PC Linux | *3 | - | *3 | - | *4 | - |
| Jetson Nano | *3 | - | *3 | - | *4 | - |
| Raspberry Pi 2/4 | - | | | | | |

*1 Enable network discovery and file sharing when using a Windows account without administrative privileges
*2 When Firewall is ON, allow connections by applications in the following way:
Open Firewall Options (System Preferences > Security & Privacy > Firewall > Firewall Options...)
Set "Allow incoming connections" for the applications
*3 Set the network setting to "Link Local Only"
*4 Set the network setting to "Automatic (DHCP)"

Camera Remote SDK uses the following ports for such as searching the connected cameras.
If Firewall is ON, the camera may not be recognized. Try one of the followings:
- Register your application which using Camera Remote SDK as an exception to Firewall.
- Change the configuration of the ports as follows to enable communications.
  Remote port
    UDP port: 1900, 32768 - 61000
    TCP port: 80, 8080, 22, 64321, 15740
  Local port
    UDP port: 1900, 49152 - 65535
    TCP port: 49152 - 65535

Also because of the above, please note that there is a possibility security software makes Warning if your application has no digital signature.

## Pairing

- First, select “Network > Transfer/Remote > PC Remote Function > Pairing” to display the pairing standby. Then call the Connect() function from your application.
  Then, the camera will change to the pairing confirmation screen. Select OK.

# Camera body settings for wired LAN connection by SSH

When connecting to an SSH support models, the following settings must be made on the camera itself.

ILME-FX6 is used as an example here. For other models, refer to the Help Guide.

### MENU > Network > Access Authentication

- Decide a User name and Password
  In the “User Name” and “Input Password” fields, enter the User name and Password used to connect to the host device (PC, smartphone, tablet, etc.).
  Please refer to each help guide and check if necessary.
- Fingerprint confirmation
  When remote operating a camera that requires SSH authentication, make sure that the user has a correct fingerprint before allowing the connection. You can see the fingerprint generated by the camera body in Show Settings> Fingerprint.

### MENU > Network > Wired LAN

- Please set “Setting” to “On”.
- Please set “Cam. Remote Ctrl” to “Enable”.

### MENU > Network > Wired LAN > Detail Settings

- The default setting for “DHCP” is “On”. If the camera is connecting to a router with a DHCP service, set the setting to "On" to automatically assign an IP address. If you want to use a network HUB or connect directly to the host PC, change the setting to "Off" and set the IP address manually.
- Refer to the table of MENU> Network> Wired LAN> IP Address Setting on the previous page for the combination of router use and hub use and DHCP setting. Replace "Auto" with "On" and "Manual" with "Off" to read.
- The following TCP ports are used for communication with cameras that require SSH authentication.

| | Remote port | Local port | Description |
|---|---|---|---|
| SSH | 22 | - | Used for SSH connection to the camera. |
| HTTP | 8080 | 58081 - 58207 | It can be used by users to access content in the slot's media.<br>Increases each time Connect() is called.<br>Rotation with 127.<br>See “Get the MediaProfile”<br><br>Note that when multiple SSH-authenticated cameras are connected at the same time, users will be communicating with localhost instead of the camera's IP address.<br>Ex.) Cam1(192.168.10.3) = localhost:58081<br>Cam2(192.168.10.4) = localhost:58082 |

# Uninstallation

## Delete all related folders and files.

When uninstalling your application which uses Camera Remote SDK, delete the following folders and files , or delete the information stored in the files with EditSDKInfo.

OS common :
..\CrAdapter*.*
..*.*
Win :
..\Users\<User Name>\AppData\Roaming\Camera Remote SDK*.*
Mac :
../Users/<User Name>/Library/Preferences/Camera Remote SDK/*.*
Linux :
../home/<User Name>/Camera Remote SDK/*.*

# API list

Whether or not each API can be used is determined according to the SDK control mode. The Mode column indicates the availability of RemoteControlMode and ContentsTransferMode in “R” and “C”.
The ContentsTransferMode feature was added in version 1.05.00.
Be sure to check Enable Status for APIs that have Enable Status. Examples are DownloadSettingFile and Camera-Setting Save Operation Enable Status, RequestLensInformation and Lens Information Enable Status. Operation cannot be guaranteed if executed in the Disable state.



| No. | APIs | Outline | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Init | Initialize the Camera Remote SDK for use. | - | Basic API of the SDK that can always be executed | | | | | | | | | | | | |
| 2 | Release | Terminate the Camera Remote SDK. | - | Basic API of the SDK that can always be executed | | | | | | | | | | | | |
| 3 | GetSDKVersion | Get SDK version number. | - | Basic API of the SDK that can always be executed | | | | | | | | | | | | |
| 4 | GetSDKSerial | Get SDK serial number. | - | Basic API of the SDK that can always be executed | | | | | | | | | | | | |
| 5 | EditSDKInfo | Edit the information about the SDK stored in the config file. | - | Basic API of the SDK that can always be executed | | | | | | | | | | | | |
| 6 | EnumCameraObjects | Make a list of corresponding camera for the Camera Remote SDK. | - | ✓ | | | | | | | | | | | | |
| 7 | CreateCameraObjectInfo | Create an ICrCameraObjectInfo object represents a Camera. | - | ✓ | | | | | | | | | | | | |
| 8 | CreateCameraObjectInfoUSBConnection | Create an ICrCameraObjectInfo object that represents a camera to be connecting via USB. | R/C | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 9 | CreateCameraObjectInfoEthernetConnection | Create an ICrCameraObjectInfo object that represents a camera to be connecting via Ethernet. | R/C | ✓ | ✓ | - | - | - | - | - | - | ✓ | - | - | - | - |
| 10 | GetFingerprint | Get the fingerprint of the camera to connect with SSH authentication. | R/C | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 11 | Connect | Connect to a Camera using a ICrCameraObjectInfo object before manipulation. | R/C | ✓ | | | | | | | | | | | | |
| 12 | Disconnect | Disconnect from the Camera after use. | R/C | ✓ | | | | | | | | | | | | |
| 13 | ReleaseDevice | Remove resources allocated with the Connect function. | R/C | ✓ | | | | | | | | | | | | |
| 14 | GetDeviceProperties | Read camera settings. | R/C | ✓ | | | | | | | | | | | | |
| 15 | ReleaseDeviceProperties | Release the CrDeviceProperty objects allocated by GetDeviceProperties. | R/C | ✓ | | | | | | | | | | | | |
| 16 | SetDeviceProperty | Set camera settings. | R | ✓ | | | | | | | | | | | | |
| 17 | SendCommand | Send control command. | R/C | ✓ | | | | | | | | | | | | |
| 18 | GetLiveViewImage | Read the latest live-view image data from the Camera into the memory of the current machine. | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 19 | GetLiveViewImageInfo | This function returns the size of the live-view image. | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 20 | GetLiveViewProperties | Get live view properties from the camera. | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 21 | ReleaseLiveViewProperties | Release the CrLiveViewProperty objects allocated by GetLiveViewProperties. | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |

(2/2)

| No. | APIs | Outline | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | GetDeviceSetting | This function returns the value of settings in the Camera Remote SDK. | R/C | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 23 | SetDeviceSetting | This function modifies the value of settings in the Camera Remote SDK. | R/C | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| | | The input parameter “Setting_Key_PartialBuffer” is available. | R | *2 | *2 | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 24 | SetSaveInfo | This function modifies settings for saving pictures. | R/C | | | | | | | ✓ | | | | | | |
| 25 | GetSelectDeviceProperties | Specify and read the device property from the camera. | R/C | | | | | | | ✓ | | | | | | |
| 26 | GetSelectLiveViewProperties | Specify and read the live view property from the camera. | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 27 | GetDateFolderList | Get date folder. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 28 | GetContentsHandleList | Get content handle array in the date folder. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 29 | GetContentsDetailInfo | Get contents Information. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 30 | ReleaseDateFolderList | Release the CrMtpFolderInfo objects allocated by GetDateFolderList. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 31 | ReleaseContentsHandleList | Release the CrMtpContentsInfo object allocated by GetContentsHandleList. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 32 | PullContentsFile (*1) | Get(download) contents file. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 33 | GetContentsThumbnailImage | Read thumbnail image data into the memory of the current machine. | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 34 | DownloadSettingFile | Get(download) the camera settings file. | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 35 | UploadSettingFile | Update(upload) the camera settings file. | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 36 | RequestDisplayStringList | Request a list of display menu strings. | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 37 | GetDisplayStringTypes | Get referenceable display menu string type. | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 38 | GetDisplayStringList | Get the list of display menu strings. | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 39 | ReleaseDisplayStringTypes | Release the CrDisplayStringType objects allocated by GetDisplayStringTypes. | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 40 | ReleaseDisplayStringList | Release the CrDisplayStringListInfo objects allocated by GetDisplayStringList. | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 41 | GetMediaProfile | Get the MediaProfile Lists. | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 42 | ReleaseMediaProfile | Release the MediaProfile Lists. | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 43 | RequestLensInformation | Request the acquisition of Lens information. | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 44 | GetLensInformation | Get the Lens information Lists. | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 45 | ReleaseLensInformation | Release the Lens information Lists. | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |

*1 : Large files may not be handled depending on the OS.
*2 : Only Ethernet(Wired LAN)
Note : The content transfer function cannot guarantee the transfer of content taken by other cameras.

# Function list

Please update each camera to the latest System Software (Firmware) before use.
Whether or not each Functions can be used is determined according to the SDK control mode. The Mode column indicates the availability of RemoteControlMode and ContentsTransferMode in “R” and “C”. The ContentsTransferMode feature was added in version 1.05.00.
Be sure to check Enable Status for DeviceProperty/Command that have Enable Status. Examples are Zoom Operation and Zoom Operation Enable Status, Media Format and MediaSLOT1/2 Format Enable Status. Operation cannot be guaranteed if executed in the Disable state.



| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Shutter Half Release | CrDeviceProperty_S1 | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 2 | Shutter Release | CrCommandId_Release | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 3 | AELock Indication | CrDeviceProperty_AEL | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 4 | FEL Lock Indication | CrDeviceProperty_FEL | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 5 | AWBLock Indication | CrDeviceProperty_AWBL | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 6 | F-Number | CrDeviceProperty_FNumber | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 7 | Exposure Bias Compensation | CrDeviceProperty_ExposureBiasCompensation | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 8 | Shutter Speed | CrDeviceProperty_ShutterSpeed | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 9 | ISO Sensitivity | CrDeviceProperty_IsoSensitivity | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 10 | Focus Area | CrDeviceProperty_FocusArea | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 11 | Exposure Program Mode | CrDeviceProperty_ExposureProgramMode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 12 | Compress File Format(Still) | CrDeviceProperty_CompressionFileFormatStill | R | ✓ | - | ✓ | - | - | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 13 | File Format(Still) | CrDeviceProperty_FileType | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 14 | Media SLOT1 File Format(Still) | CrDeviceProperty_MediaSLOT1_FileType | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 15 | Media SLOT2 File Format(Still) | CrDeviceProperty_MediaSLOT2_FileType | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 16 | JPEG Quality | CrDeviceProperty_JpegQuality | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 17 | Media SLOT1 JPEG Quality | CrDeviceProperty_MediaSLOT1_JpegQuality | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 18 | Media SLOT2 JPEG Quality | CrDeviceProperty_MediaSLOT2_JpegQuality | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 19 | White Balance | CrDeviceProperty_WhiteBalance | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 20 | Focus Mode | CrDeviceProperty_FocusMode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 21 | Exposure Metering Mode | CrDeviceProperty_MeteringMode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 22 | Flash Mode | CrDeviceProperty_FlashMode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 23 | Flash Compensation | CrDeviceProperty_FlashCompensation | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 24 | Wireless Flash Setting | CrDeviceProperty_WirelessFlash | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 25 | Red Eye Reduction | CrDeviceProperty_RedEyeReduction | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| 26 | Still Capture Mode | CrDeviceProperty_DriveMode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 27 | Dynamic Range Optimizer | CrDeviceProperty_DRO | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 28 | Image Size | CrDeviceProperty_ImageSize | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 29 | Media SLOT1 Image Size | CrDeviceProperty_MediaSLOT1_ImageSize | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 30 | Media SLOT2 Image Size | CrDeviceProperty_MediaSLOT2_ImageSize | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 31 | Aspect Ratio | CrDeviceProperty_AspectRatio | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 32 | Picture Effect | CrDeviceProperty_PictureEffect | R | - | ✓ | - | ✓ | ✓ | - | - | ✓ | - | - | - | - | ✓ |
| 33 | Color Temperature | CrDeviceProperty_Colortemp | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 34 | Biaxial Fine Tuning A-B Direction | CrDeviceProperty_ColorTuningAB | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 35 | Biaxial Fine Tuning G-M Direction | CrDeviceProperty_ColorTuningGM | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 36 | Live View Display Effect | CrDeviceProperty_LiveViewDisplayEffect | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 37 | Still Image Save Destination | CrDeviceProperty_StillImageStoreDestination | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 38 | Position Key Setting | CrDeviceProperty_PriorityKeySettings | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | - | - | ✓ | ✓ |
| 39 | Focus Magnifier Setting | CrDeviceProperty_Focus_Magnifier_Setting | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 40 | Date/Time Setting | CrDeviceProperty_DateTime_Settings | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 41 | Focus Near/Far Setting | CrDeviceProperty_NearFar | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 42 | Live View Image Quality | CrDeviceProperty_LiveView_Image_Quality | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 43 | Interval REC Mode | CrDeviceProperty_Interval_Rec_Mode | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 44 | Still Image Trans Size | CrDeviceProperty_Still_Image_Trans_Size | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 45 | RAW+J PC Save Image | CrDeviceProperty_RAW_J_PC_Save_Image | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 46 | Custom WB Capture Standby | CrDeviceProperty_CustomWB_Capture_Standby | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 47 | Custom WB Capture Standby Cancel | CrDeviceProperty_CustomWB_Capture_Standby_Cancel | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 48 | Custom WB Capture | CrDeviceProperty_CustomWB_Capture | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 49 | Shooting File Info | CrDeviceProperty_SnapshotInfo | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | Battery Remaining | CrDeviceProperty_BatteryRemain | R/C | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 51 | Battery Level Indicator | CrDeviceProperty_BatteryLevel | R/C | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 52 | Movie Recording State | CrDeviceProperty_RecordingState | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 53 | LiveView Status | CrDeviceProperty_LiveViewStatus | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 54 | Focus Indication | CrDeviceProperty_FocusIndication | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 55 | Media SLOT1 Status | CrDeviceProperty_MediaSLOT1_Status | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 56 | Media SLOT1 Remaining number shots | CrDeviceProperty_MediaSLOT1_RemainingNumber | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 57 | Media SLOT1 Remaining shooting time | CrDeviceProperty_MediaSLOT1_RemainingTime | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 58 | Media SLOT1 Full Format Enable Status | CrDeviceProperty_MediaSLOT1_FormatEnableStatus | R | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 59 | Media SLOT1 Quick Format Enable Status | CrDeviceProperty_MediaSLOT1_QuickFormatEnableStatus | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 60 | Media SLOT2 Status | CrDeviceProperty_MediaSLOT2_Status | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | - |
| 61 | Media SLOT2 Remaining number shots | CrDeviceProperty_MediaSLOT2_RemainingNumber | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | - | - |
| 62 | Media SLOT2 Remaining shooting time | CrDeviceProperty_MediaSLOT2_RemainingTime | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | - |
| 63 | Media SLOT2 Full Format Enable Status | CrDeviceProperty_MediaSLOT2_FormatEnableStatus | R | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | - | - | ✓ | ✓ | - | - |
| 64 | Media SLOT2 Quick Format Enable Status | CrDeviceProperty_MediaSLOT2_QuickFormatEnableStatus | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 65 | Media Format Progress Rate | CrDeviceProperty_Media_FormatProgressRate | R | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 66 | Execute Format the Media | CrCommandId_MediaFormat | R | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 67 | Execute Quick Format the Media | CrCommandId_MediaQuickFormat | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 68 | AF Area Position | CrDeviceProperty_AF_Area_Position | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 69 | Zoom Scale | CrDeviceProperty_Zoom_Scale | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 70 | Zoom Setting | CrDeviceProperty_Zoom_Setting | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 71 | Zoom Operation | CrDeviceProperty_Zoom_Operation | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 72 | File Format(Movie) | CrDeviceProperty_Movie_File_Format | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 73 | Recording Setting(Movie) | CrDeviceProperty_Movie_Recording_Setting | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 74 | Recording Frame Rate Setting(Movie) | CrDeviceProperty_Movie_Recording_FrameRateSetting | R | ✓ | - | ✓ | - | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 75 | Interval REC Status | CrDeviceProperty_Interval_Rec_Status | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 76 | Control Movie Rec button | CrCommandId_MovieRecord | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 77 | Custom WB Execution State | CrDeviceProperty_CustomWB_Execution_State | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 78 | Custom WB Capturable Area | CrDeviceProperty_CustomWB_Capturable_Area | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 79 | Custom WB Capture Frame Size | CrDeviceProperty_CustomWB_Capture_Frame_Size | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 80 | Custom WB Capture Operation Enable Status | CrDeviceProperty_CustomWB_Capture_Operation | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 81 | Zoom Operation Enable Status | CrDeviceProperty_Zoom_Operation_Status | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 82 | Zoom Bar Information | CrDeviceProperty_Zoom_Bar_Information | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 83 | Zoom Type Status | CrDeviceProperty_Zoom_Type_Status | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 84 | RAW File Compression Type | CrDeviceProperty_RAW_FileCompressionType | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | ✓ | - |
| 85 | Media SLOT1 RAW File Compression Type | CrDeviceProperty_MediaSLOT1_RAW_FileCompressionType | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | - | - |
| 86 | Media SLOT2 RAW File Compression Type | CrDeviceProperty_MediaSLOT2_RAW_FileCompressionType | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | - | - |
| 87 | Cancel Media Format Enable Status | CrDeviceProperty_Cancel_Media_FormatEnableStatus | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 88 | Cancel Media Format | CrCommandId_CancelMediaFormat | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 89 | Shutter Half Release and Release | CrCommandId_S1andRelease | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 90 | Save Zoom&FocusPosition in presets | CrDeviceProperty_ZoomAndFocusPosition_Save | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 91 | Load Zoom&FocusPosition from presets | CrDeviceProperty_ZoomAndFocusPosition_Load | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 92 | Remocon Zoom Speed Type | CrDeviceProperty_Remocon_Zoom_Speed_Type | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 93 | Zoom Speed Range | CrDeviceProperty_Zoom_Speed_Range | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | - |
| 94 | Sdk Control Mode | CrDeviceProperty_SdkControlMode | R/C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 95 | Get content accessibility status | CrDeviceProperty_ContentsTransferStatus | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 96 | Cancel Content transfer Enable Status | CrDeviceProperty_ContentsTransferCancelEnableStatus | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 97 | Content transfer Progress | CrDeviceProperty_ContentsTransferProgress | C | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 98 | Cancel Contents transfer | CrCommandId_CancelContentsTransfer | R | ✓ | - | ✓ | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 99 | Iris Mode Setting | CrDeviceProperty_IrisModeSetting | R | - | - | ✓ | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 100 | Shutter Mode Setting | CrDeviceProperty_ShutterModeSetting | R | - | - | ✓ | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 101 | Gain Control Setting | CrDeviceProperty_GainControlSetting | R | - | - | ✓ | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 102 | Gain Base Iso Sensitivity | CrDeviceProperty_GainBaseIsoSensitivity | R | - | - | - | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 103 | Gain Base Sensitivity | CrDeviceProperty_GainBaseSensitivity | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 104 | Exposure Index | CrDeviceProperty_ExposureIndex | R | - | - | - | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 105 | BaseLook Value | CrDeviceProperty_BaseLookValue | R | - | - | - | - | - | - | - | - | ✓ | ✓ | ✓ | - | - |
| 106 | Playback Media | CrDeviceProperty_PlaybackMedia | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | - | - |
| 107 | Monitor DISP(Screen Display) Mode Candidate | CrDeviceProperty_DispModeCandidate | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 108 | Monitor DISP(Screen Display) Mode Setting | CrDeviceProperty_DispModeSetting | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 109 | Monitor DISP(Screen Display) Mode | CrDeviceProperty_DispMode | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 110 | Touch Operation | CrDeviceProperty_TouchOperation | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 111 | Select Finder/Monitor | CrDeviceProperty_SelectFinder | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | - | - |
| 112 | Auto Power OFF Temperature | CrDeviceProperty_AutoPowerOffTemperature | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 113 | Body Key Lock | CrDeviceProperty_BodyKeyLock | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 114 | Image ID(Numerical Value) Setting | CrDeviceProperty_ImageID_Num_Setting | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 115 | Image ID(Numerical Value) | CrDeviceProperty_ImageID_Num | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 116 | Image ID(String) | CrDeviceProperty_ImageID_String | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 117 | Exposure Control Mode | CrDeviceProperty_ExposureCtrlType | R | - | - | ✓ | - | - | - | - | - | - | ✓ | ✓ | - | - |
| 118 | Monitor LUT Setting(All Line) | CrDeviceProperty_MonitorLUTSetting | R | - | - | - | - | - | - | - | - | - | ✓ | ✓ | - | - |
| 119 | ISO Current Sensitivity | CrDeviceProperty_IsoCurrentSensitivity | R | - | - | ✓ | - | - | - | - | - | ✓ | ✓ | ✓ | ✓ | - |
| 120 | Camera-Setting Save Operation Enable Status | CrDeviceProperty_CameraSetting_SaveOperationEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 121 | Camera-Setting Read Operation Enable Status | CrDeviceProperty_CameraSetting_ReadOperationEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 122 | Camera-Setting Save/Read State | CrDeviceProperty_CameraSetting_SaveRead_State | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 123 | Camera Setting Reset Enable Status | CrDeviceProperty_CameraSettingsResetEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 124 | Execute Camera Setting Reset | CrCommandId_CameraSettingsReset | R | - | - | ✓ | - | - | - | ✓ | - | - | ✓ | ✓ | ✓ | - |
| 125 | APS-C or Full Switching Setting | CrDeviceProperty_APS_C_or_Full_SwitchingSetting | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | ✓ | - |
| 126 | APS-C or Full Switching Enable Status | CrDeviceProperty_APS_C_or_Full_SwitchingEnableStatus | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | ✓ | - |
| 127 | Execute APS-C or Full Switching | CrCommandId_APS_C_or_Full_Switching | R | ✓ | - | ✓ | - | - | - | ✓ | - | - | ✓ | - | ✓ | - |
| 128 | Execute Movie Rec Button (2nd) | CrCommandId_MovieRecButtonToggle | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 129 | Execute Cancel Remote Touch Operation | CrCommandId_CancelRemoteTouchOperation | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 130 | Focal Distance in Meter | CrDeviceProperty_FocalDistanceInMeter | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 131 | Focal Distance in Feet | CrDeviceProperty_FocalDistanceInFeet | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 132 | Focal Distance Unit Setting | CrDeviceProperty_FocalDistanceUnitSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 133 | Digital Zoom Scale | CrDeviceProperty_DigitalZoomScale | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 134 | Zoom Distance | CrDeviceProperty_ZoomDistance | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 135 | Zoom Distance Unit Setting | CrDeviceProperty_ZoomDistanceUnitSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 136 | Shutter Mode Status | CrDeviceProperty_ShutterModeStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 137 | Shutter Slow | CrDeviceProperty_ShutterSlow | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 138 | Shutter Slow Frames | CrDeviceProperty_ShutterSlowFrames | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 139 | Recording Resolution For Main(Movie) | CrDeviceProperty_Movie_Recording_ResolutionForMain | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 140 | Recording Resolution For Proxy(Movie) | CrDeviceProperty_Movie_Recording_ResolutionForProxy | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 141 | Recording Frame Rate Proxy Setting(Movie) | CrDeviceProperty_Movie_Recording_FrameRateProxySetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 142 | Movie Shooting Mode | CrDeviceProperty_MovieShootingMode | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 143 | Movie Shooting Mode Color Gamut | CrDeviceProperty_MovieShootingModeColorGamut | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 144 | Movie Shooting Mode Target Display | CrDeviceProperty_MovieShootingModeTargetDisplay | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 145 | Depth of Field Adjustment Mode | CrDeviceProperty_DepthOfFieldAdjustmentMode | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 146 | Depth of Field Adjustment Interlocking Mode State | CrDeviceProperty_DepthOfFieldAdjustmentInterlockingMode | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 147 | Color Temperature Step | CrDeviceProperty_ColortempStep | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 148 | White Balance Mode Setting | CrDeviceProperty_WhiteBalanceModeSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 149 | White Balance Tint | CrDeviceProperty_WhiteBalanceTint | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 150 | White Balance Tint Step | CrDeviceProperty_WhiteBalanceTintStep | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 151 | Execute the Focus Operation | CrDeviceProperty_Focus_Operation | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 152 | Focus Speed Range | CrDeviceProperty_Focus_Speed_Range | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 153 | Shutter ECS Setting | CrDeviceProperty_ShutterECSSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 154 | Shutter ECS Number | CrDeviceProperty_ShutterECSNumber | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 155 | Shutter ECS Number Step | CrDeviceProperty_ShutterECSNumberStep | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 156 | Shutter ECS Frequency | CrDeviceProperty_ShutterECSFrequency | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 157 | Button Assignment Assignable 1 | CrDeviceProperty_ButtonAssignmentAssignable1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 158 | Button Assignment Assignable 2 | CrDeviceProperty_ButtonAssignmentAssignable2 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 159 | Button Assignment Assignable 3 | CrDeviceProperty_ButtonAssignmentAssignable3 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 160 | Button Assignment Assignable 4 | CrDeviceProperty_ButtonAssignmentAssignable4 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 161 | Button Assignment Assignable 5 | CrDeviceProperty_ButtonAssignmentAssignable5 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 162 | Button Assignment Assignable 6 | CrDeviceProperty_ButtonAssignmentAssignable6 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 163 | Button Assignment Assignable 7 | CrDeviceProperty_ButtonAssignmentAssignable7 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 164 | Button Assignment Assignable 8 | CrDeviceProperty_ButtonAssignmentAssignable8 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 165 | Button Assignment Assignable 9 | CrDeviceProperty_ButtonAssignmentAssignable9 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 166 | Button Assignment LensAssignable 1 | CrDeviceProperty_ButtonAssignmentLensAssignable1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 167 | Assignable Button 1 | CrDeviceProperty_AssignableButton1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 168 | Assignable Button 2 | CrDeviceProperty_AssignableButton2 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 169 | Assignable Button 3 | CrDeviceProperty_AssignableButton3 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 170 | Assignable Button 4 | CrDeviceProperty_AssignableButton4 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 171 | Assignable Button 5 | CrDeviceProperty_AssignableButton5 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 172 | Assignable Button 6 | CrDeviceProperty_AssignableButton6 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 173 | Assignable Button 7 | CrDeviceProperty_AssignableButton7 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 174 | Assignable Button 8 | CrDeviceProperty_AssignableButton8 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 175 | Assignable Button 9 | CrDeviceProperty_AssignableButton9 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 176 | LensAssignable Button 1 | CrDeviceProperty_LensAssignableButton1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 177 | Focus Mode Setting | CrDeviceProperty_FocusModeSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 178 | Shutter Angle | CrDeviceProperty_ShutterAngle | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 179 | Shutter Setting | CrDeviceProperty_ShutterSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 180 | Shutter Mode | CrDeviceProperty_ShutterMode | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 181 | Shutter Speed Value | CrDeviceProperty_ShutterSpeedValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 182 | Shutter Speed Current Value | CrDeviceProperty_ShutterSpeedCurrentValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 183 | ND Filter | CrDeviceProperty_NDFilter | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 184 | ND Filter Mode | CrDeviceProperty_NDFilterMode | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 185 | ND Filter Mode Setting | CrDeviceProperty_NDFilterModeSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 186 | ND Filter Value | CrDeviceProperty_NDFilterValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 187 | Gain Unit Setting | CrDeviceProperty_GainUnitSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 188 | Gain dB Value | CrDeviceProperty_GaindBValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 189 | Gain dB Current Value | CrDeviceProperty_GaindBCurrentValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 190 | AWB | CrDeviceProperty_AWB | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 191 | SceneFile Index | CrDeviceProperty_SceneFileIndex | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 192 | Current SceneFile Edited Info | CrDeviceProperty_CurrentSceneFileEdited | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 193 | Movie Play button | CrDeviceProperty_MoviePlayButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 194 | Movie Play Pause button | CrDeviceProperty_MoviePlayPauseButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 195 | Movie Play Stop button | CrDeviceProperty_MoviePlayStopButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 196 | Movie Forward button | CrDeviceProperty_MovieForwardButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 197 | Movie Rewind button | CrDeviceProperty_MovieRewindButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 198 | Movie Next button | CrDeviceProperty_MovieNextButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 199 | Movie Prev button | CrDeviceProperty_MoviePrevButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 200 | Movie RecReview button | CrDeviceProperty_MovieRecReviewButton | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 201 | Face Eye Detection AF | CrDeviceProperty_FaceEyeDetectionAF | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 202 | AF Transition Speed | CrDeviceProperty_AFTransitionSpeed | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 203 | AF Subj Shift Sens | CrDeviceProperty_AFSubjShiftSens | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 204 | AF Assist | CrDeviceProperty_AFAssist | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 205 | ND PRESET or VARIABLE Switching Setting | CrDeviceProperty_NDPresetOrVariableSwitchingSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 206 | Function of Remote Touch Operation | CrDeviceProperty_FunctionOfRemoteTouchOperation | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 207 | Execute Remote Touch Operation(x,y) | CrDeviceProperty_RemoteTouchOperation | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 208 | Movie Playing State | CrDeviceProperty_MoviePlayingState | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 209 | Movie Playing Speed | CrDeviceProperty_MoviePlayingSpeed | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 210 | Media SLOT1 Player | CrDeviceProperty_MediaSLOT1Player | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 211 | Media SLOT2 Player | CrDeviceProperty_MediaSLOT2Player | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 212 | Battery Remain Display Unit | CrDeviceProperty_BatteryRemainDisplayUnit | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 213 | Battery Remaining in minutes | CrDeviceProperty_BatteryRemainingInMinutes | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 214 | Battery Remaining in voltage | CrDeviceProperty_BatteryRemainingInVoltage | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 215 | Power Source | CrDeviceProperty_PowerSource | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 216 | DC voltage | CrDeviceProperty_DCVoltage | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 217 | Focus TouchSpot Status | CrDeviceProperty_FocusTouchSpotStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 218 | Focus Tracking Status | CrDeviceProperty_FocusTrackingStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 219 | Recorder Clip Name Create by The Next Rec. | CrDeviceProperty_RecorderClipName | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 220 | Recorder Control Main Setting | CrDeviceProperty_RecorderControlMainSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 221 | Recorder Control Proxy Setting | CrDeviceProperty_RecorderControlProxySetting | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | - | - | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 222 | Recorder Start Main | CrDeviceProperty_RecorderStartMain | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 223 | Recorder Start Proxy | CrDeviceProperty_RecorderStartProxy | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 224 | Recorder Main Status | CrDeviceProperty_RecorderMainStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 225 | Recorder Proxy Status | CrDeviceProperty_RecorderProxyStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 226 | Recorder Ext Raw Status | CrDeviceProperty_RecorderExtRawStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 227 | Information of Recorder Save Destination | CrDeviceProperty_RecorderSaveDestination | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 228 | Assignable Button Indicator 1 | CrDeviceProperty_AssignableButtonIndicator1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 229 | Assignable Button Indicator 2 | CrDeviceProperty_AssignableButtonIndicator2 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 230 | Assignable Button Indicator 3 | CrDeviceProperty_AssignableButtonIndicator3 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 231 | Assignable Button Indicator 4 | CrDeviceProperty_AssignableButtonIndicator4 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 232 | Assignable Button Indicator 5 | CrDeviceProperty_AssignableButtonIndicator5 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 233 | Assignable Button Indicator 6 | CrDeviceProperty_AssignableButtonIndicator6 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 234 | Assignable Button Indicator 7 | CrDeviceProperty_AssignableButtonIndicator7 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 235 | Assignable Button Indicator 8 | CrDeviceProperty_AssignableButtonIndicator8 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 236 | Assignable Button Indicator 9 | CrDeviceProperty_AssignableButtonIndicator9 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 237 | LensAssignable Button Indicator 1 | CrDeviceProperty_LensAssignableButtonIndicator1 | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 238 | Software Version | CrDeviceProperty_SoftwareVersion | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | - | - | ✓ | - |
| 239 | Movie Rec Button (2nd) Enable Status | CrDeviceProperty_MovieRecButtonToggleEnableStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 240 | Remote Touch Operation Enable Status | CrDeviceProperty_RemoteTouchOperationEnableStatus | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 241 | Cancel Remote Touch Operation Enable Status | CrDeviceProperty_CancelRemoteTouchOperationEnableStatus | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| 242 | Lens Information Enable Status | CrDeviceProperty_LensInformationEnableStatus | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 243 | Follow Focus Position | CrDeviceProperty_FollowFocusPositionSetting | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 244 | Follow Focus Position Current Value | CrDeviceProperty_FollowFocusPositionCurrentValue | R | - | - | - | - | - | - | - | - | ✓ | - | - | - | - |
| 245 | Focus Bracket Shot Number | CrDeviceProperty_FocusBracketShotNumber | R | - | - | ✓ | - | - | - | - | - | - | - | - | ✓ | - |
| 246 | Focus Bracket Focus Range | CrDeviceProperty_FocusBracketFocusRange | R | - | - | ✓ | - | - | - | - | - | - | - | - | ✓ | - |
| 247 | Focus Bracket Shooting Status | CrDeviceProperty_FocusBracketShootingStatus | R | - | - | ✓ | - | - | - | - | - | - | - | - | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 248 | Function of Touch Operation | CrDeviceProperty_FunctionOfTouchOperation | R | ✓ | ✓ | ✓ | - | - | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | - |
| 249 | Proxy File Format(Movie) | CrDeviceProperty_Movie_ProxyFileFormat | R | - | - | ✓ | - | - | - | ✓ | - | ✓ | - | - | ✓ | - |
| 250 | Extended Interface Mode | CrDeviceProperty_ExtendedInterfaceMode | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 251 | S&Q Frame Rate | CrDeviceProperty_SQFrameRate | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 252 | S&Q Recording Frame Rate Setting | CrDeviceProperty_SQRecordingFrameRateSetting | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 253 | S&Q Recording Setting | CrDeviceProperty_SQRecordingSetting | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 254 | Audio Recording | CrDeviceProperty_AudioRecording | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 255 | Audio Input Master Level | CrDeviceProperty_AudioInputMasterLevel | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 256 | Time Code Preset | CrDeviceProperty_TimeCodePreset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 257 | Time Code Format | CrDeviceProperty_TimeCodeFormat | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 258 | Time Code Run | CrDeviceProperty_TimeCodeRun | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 259 | Time Code Make | CrDeviceProperty_TimeCodeMake | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 260 | User Bit Preset | CrDeviceProperty_UserBitPreset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 261 | User Bit Time Rec | CrDeviceProperty_UserBitTimeRec | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 262 | Image Stabilization Steady Shot | CrDeviceProperty_ImageStabilizationSteadyShot | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 263 | Image Stabilization Steady Shot(Movie) | CrDeviceProperty_Movie_ImageStabilizationSteadyShot | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 264 | Silent Mode | CrDeviceProperty_SilentMode | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 265 | Silent Mode Aperture Drive in AF | CrDeviceProperty_SilentModeApertureDriveInAF | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 266 | Silent Mode Shutter When Power Off | CrDeviceProperty_SilentModeShutterWhenPowerOff | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | - | - |
| 267 | Silent Mode Auto Pixel Mapping | CrDeviceProperty_SilentModeAutoPixelMapping | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | - | - |
| 268 | Shutter Type | CrDeviceProperty_ShutterType | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | - | - |
| 269 | Picture Profile | CrDeviceProperty_PictureProfile | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 270 | Picture Profile Black Level | CrDeviceProperty_PictureProfile_BlackLevel | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 271 | Picture Profile Gamma | CrDeviceProperty_PictureProfile_Gamma | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 272 | Picture Profile Black Gamma Range | CrDeviceProperty_PictureProfile_BlackGammaRange | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 273 | Picture Profile Black Gamma Level | CrDeviceProperty_PictureProfile_BlackGammaLevel | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |

| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 274 | Picture Profile Knee Mode | CrDeviceProperty_PictureProfile_KneeMode | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 275 | Picture Profile Knee AutoSet MaxPoint | CrDeviceProperty_PictureProfile_KneeAutoSet_MaxPoint | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 276 | Picture Profile Knee AutoSet Sensitivity | CrDeviceProperty_PictureProfile_KneeAutoSet_Sensitivity | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 277 | Picture Profile Knee ManualSet Point | CrDeviceProperty_PictureProfile_KneeManualSet_Point | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 278 | Picture Profile Knee ManualSet Slope | CrDeviceProperty_PictureProfile_KneeManualSet_Slope | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 279 | Picture Profile Color Mode | CrDeviceProperty_PictureProfile_ColorMode | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 280 | Picture Profile Saturation | CrDeviceProperty_PictureProfile_Saturation | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 281 | Picture Profile Color Phase | CrDeviceProperty_PictureProfile_ColorPhase | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 282 | Picture Profile Color Depth Red | CrDeviceProperty_PictureProfile_ColorDepthRed | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 283 | Picture Profile Color Depth Green | CrDeviceProperty_PictureProfile_ColorDepthGreen | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 284 | Picture Profile Color Depth Blue | CrDeviceProperty_PictureProfile_ColorDepthBlue | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 285 | Picture Profile Color Depth Cyan | CrDeviceProperty_PictureProfile_ColorDepthCyan | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 286 | Picture Profile Color Depth Magenta | CrDeviceProperty_PictureProfile_ColorDepthMagenta | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 287 | Picture Profile Color Depth Yellow | CrDeviceProperty_PictureProfile_ColorDepthYellow | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 288 | Picture Profile Detail Level | CrDeviceProperty_PictureProfile_DetailLevel | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 289 | Picture Profile Detail Adjust Mode | CrDeviceProperty_PictureProfile_DetailAdjustMode | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 290 | Picture Profile Detail Adjust V/H Balance | CrDeviceProperty_PictureProfile_DetailAdjustVHBalance | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 291 | Picture Profile Detail Adjust B/W Balance | CrDeviceProperty_PictureProfile_DetailAdjustBWBalance | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 292 | Picture Profile Detail Adjust Limit | CrDeviceProperty_PictureProfile_DetailAdjustLimit | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 293 | Picture Profile Detail Adjust Crispening | CrDeviceProperty_PictureProfile_DetailAdjustCrispening | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 294 | Picture Profile Detail Adjust Hi-Light Detail | CrDeviceProperty_PictureProfile_DetailAdjustHiLightDetail | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 295 | Copy Picture Profile | CrDeviceProperty_PictureProfile_Copy | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 296 | Picture Profile Reset Enable Status | CrDeviceProperty_PictureProfileResetEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 297 | Execute Picture Profile Reset | CrCommandId_PictureProfileReset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 298 | Creative Look | CrDeviceProperty_CreativeLook | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 299 | Creative Look Contrast | CrDeviceProperty_CreativeLook_Contrast | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |

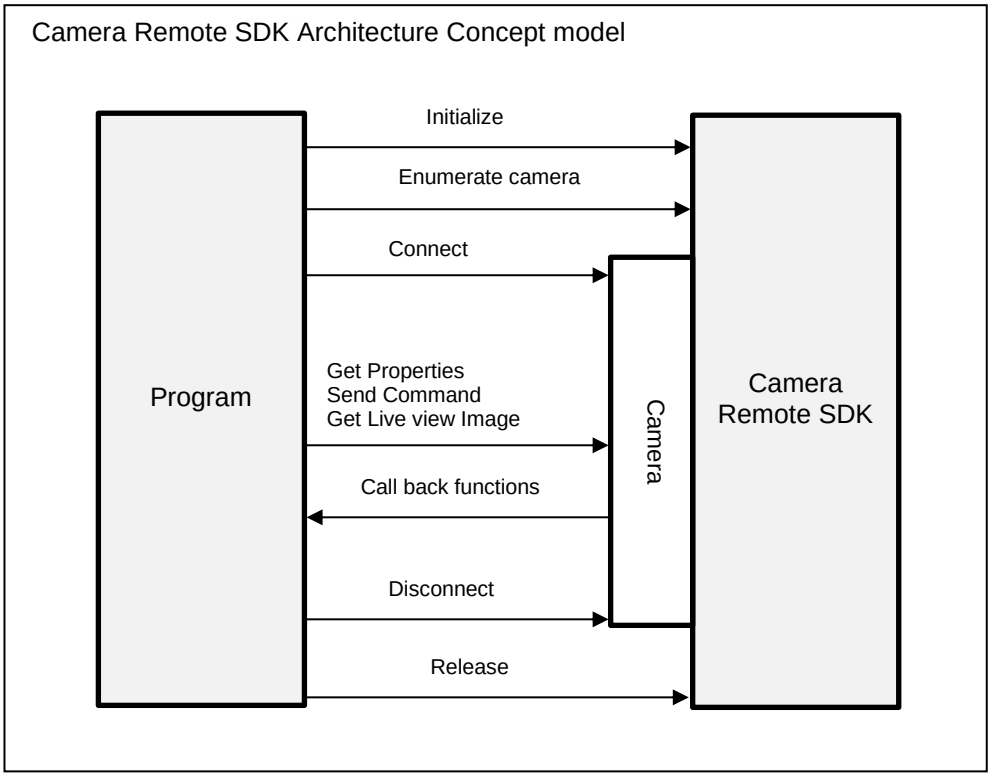
| No. | Functions | DeviceProperty Code / Command Id | Mode | ILCE-1 | ILCE-9M2 | ILCE-7RM5 | ILCE-7RM4A | ILCE-7RM4 | ILCE-7SM3 | ILCE-7M4 | ILCE-7C | ILME-FX6 | ILME-FX3 | ILME-FX30 | ZV-E1 | DSC-RX0M2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 300 | Creative Look Highlights | CrDeviceProperty_CreativeLook_Highlights | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 301 | Creative Look Shadows | CrDeviceProperty_CreativeLook_Shadows | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 302 | Creative Look Fade | CrDeviceProperty_CreativeLook_Fade | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 303 | Creative Look Saturation | CrDeviceProperty_CreativeLook_Saturation | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 304 | Creative Look Sharpness | CrDeviceProperty_CreativeLook_Sharpness | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 305 | Creative Look Sharpness Range | CrDeviceProperty_CreativeLook_SharpnessRange | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 306 | Creative Look Clarity | CrDeviceProperty_CreativeLook_Clarity | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 307 | Custom Look | CrDeviceProperty_CreativeLook_CustomLook | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 308 | Creative Look Reset Enable Status | CrDeviceProperty_CreativeLookResetEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 309 | Execute Creative Look Reset | CrCommandId_CreativeLookReset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 310 | Proxy Recording Setting | CrDeviceProperty_ProxyRecordingSetting | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 311 | Interval REC(Movie) Count Down Interval Time | CrDeviceProperty_Movie_IntervalRec_CountDownIntervalTime | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 312 | Interval REC(Movie) Recording Duration | CrDeviceProperty_Movie_IntervalRec_RecordingDuration | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 313 | Pixel Mapping Enable Status | CrDeviceProperty_PixelMappingEnableStatus | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 314 | Execute Pixel Mapping | CrCommandId_PixelMapping | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |
| 315 | Time Code Preset Reset Enable Status | CrDeviceProperty_TimeCodePresetResetEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 316 | Execute Time Code Preset Reset | CrCommandId_TimeCodePresetReset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 317 | User Bit Preset Reset Enable Status | CrDeviceProperty_UserBitPresetResetEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 318 | Execute User Bit Preset Reset | CrCommandId_UserBitPresetReset | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 319 | Sensor Cleaning Enable Status | CrDeviceProperty_SensorCleaningEnableStatus | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 320 | Execute Sensor Cleaning | CrCommandId_SensorCleaning | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 321 | Lens Version Number | CrDeviceProperty_LensVersionNumber | R | - | - | ✓ | - | - | - | ✓ | - | - | - | - | ✓ | - |
| 322 | Device Overheating State | CrDeviceProperty_DeviceOverheatingState | R | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| 323 | Execute Power Off | CrCommandId_PowerOff | R | - | - | - | - | - | - | - | - | - | - | - | ✓ | - |

# Operational Flow and Sequences

This section describes the basic operational flow of Camera Remote SDK.

At the beginning of all camera operations, Init() must be called to initialize Camera Remote SDK, and at the end of the operation, Release() must be called to release all resources.

EnumCameraObjects() enumerates connected cameras that can be connected with this Camera




Remote SDK. The ICrEnumCameraObjectInfo object has the list of valid camera objects.

ICrEnumCameraObjectInfo::GetCameraObjectInfo(CrInt32 index) returns ICrCameraObjectInfo specified by the parameter “index”. With the ICrCameraObjectInfo object, call the Connect() method to connect to the camera. Note that before calling Connect(), the IDeviceCallback function object needs to be prepared. The callback functions notify the status changes of the camera and the connection. When the connection established, OnConnected() is called. When the connection is disconnected, OnDisconnected() is called. When the camera status is changed, some other callback functions are called depending on the camera status, or warning / error messages are notified by the callback functions.

Connect() returns a CrDeviceHandle. The device handle is always used to operate the camera, for example to get or change properties, to capture image, to get live view images and so on. But just calling Connect() and receiving no error is not enough to know the timing the device is connected, and if the handle is validated. After the OnConnected() callback is called, the connection is established successfully, and the device handle is valid.

After using the camera, by calling the Disconnect() method with the device handle, the disconnect process starts. Similar to the Connect() method, when the OnDisconnected() callback function is called, the connection is disconnected successfully. You can call ReleaseDevice() after you receive the OnDisconnected() call-back.

*NOTE:*

*In this Camera Remote SDK, only one camera connection is guaranteed at the same time.*

# Initialize and Release Camera Remote SDK

To initialize Camera Remote SDK, call SCRSDK::Init(0).

Init() needs one parameter, which must be zero.

In case of a memory allocation error or another fatal error, it returns false.

*Example:*

```
bool Init() {
        bool ret = SCRSDK::Init(0);
        if (!ret) {
                // code to handle the error
                return false;
        }
```

To terminate Camera Remote SDK, call SCRSDK::Release(). This function terminates all connections and releases the allocating resources. Note that the Release() function waits for the completion of the data transfer to be executed. When transferring huge amounts of data between the pc and the camera, this Release() function waits for the completion of the transfer. It is strongly recommended to call this method after confirming the disconnection of each device.

*Example:*

```
void Terminate() {
        SCRSDK::Release();
```

Currently, Release() always returns true.

# Enumerate Cameras

EnumCameraObjects() enumerates all connectable cameras that are physically connected to the PC. Returned ICrEnumCameraObjectInfo has the list of the cameras. The ICrEnumCameraObjectInfo object is created in Camera Remote SDK, if no camera is found, the returned pEnum is NULL.

The member function GetCount() of ICrEnumCameraObjectInfo returns the number of the discovered cameras and GetCameraObject(index) returns the ICrCameraObjectInfo object specified by the index parameter. Information of the discovered camera can be acquired through the object. The information varies depending on the connecting method. Connecting by USB allows you to acquire various information values (camera model name, product id, USB serial number, etc.).

To release ICrEnumCameraObjectInfo object, use the Release() function of the object.

*Example:*

```
void Enumerate() {
        CrError err = SCRSDK::EnumCameraObjects(&pEnum);
        if (pEnum == NULL) {
                // no cameras found
                return;
        }
        CrInt32u cntOfCamera = pEnum->GetCount(); // get number of cameras
        for (CrInt32u n = 0; n < cntOfCamera; n++) {
                ICrCameraObjectInfo *pobj = pEnum->GetCameraObjectInfo(n);
                // get connected camera information
        }
        :
        pEnum->Release(); // use Release() function of ICrEnumCameraObjectInfo
```

This enumeration function makes the list of “connectable” cameras. A Sony camera, which does not have PC remote control features or is not compatible with this Camera Remote SDK, is not listed. Refer to the supported model list of this Camera Remote SDK.

Note that ICrCameraObjectInfo *pobj in the sample code is the object owned by ICrEnumCameraObjectInfo. It means calling ICrEnumCameraObjectInfo::Release() frees the memory of ICrCameraObjectInfo that you get from the enumerator. It can no longer be accessed.

# Create a “Camera Object” with information known in advance

If the camera to be connected is determined in advance, you can create a “Camera object” with the specified information and use it as a parameter of Connect() without using EnumCameraObjects() of camera search function.

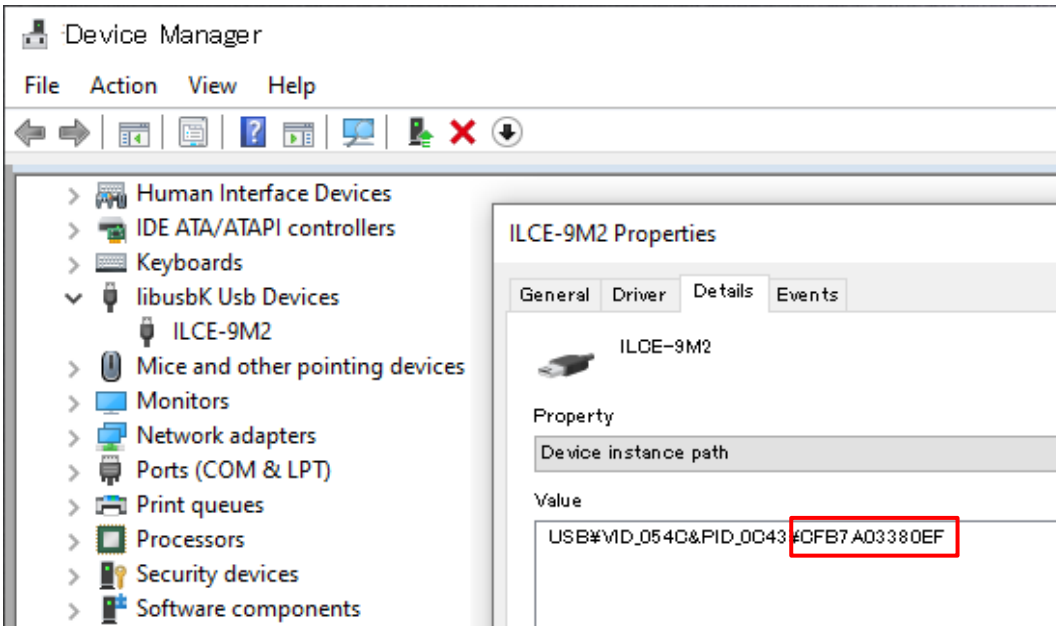
Use CreateCameraObjectInfoUSBConnection() for a USB connection and CreateCameraObjectInfoEthernetConnection() for an Ethernet connection.
The reason why we have prepared for USB connection and Ethernet connection respectively is that the required conditions differ depending on the connection method.

CreateCameraObjectInfoUSBConnection() has three parameters. The second parameter specifies the model of the camera to connect to, and the third parameter specifies the serial number of the camera to connect to. The camera object pointer created by this condition is returned with the first parameter ICrCameraObjectInfo **.
You can check the serial number of the camera by the following method.

**- Windows**

1. Connect the camera to the host PC with a USB cable and display [Device Manager]
2. Display the properties of the target camera in [Device Manager]
3. On the [Details] tab, select “Device Instance path” from the [Property] pull-down list.

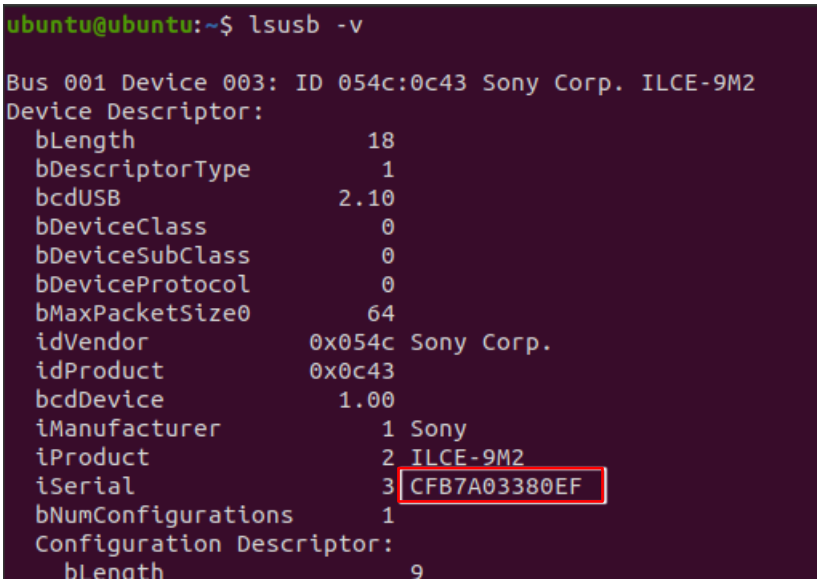
Area marked in red : USB serial number



**- Linux/RaspPi**

1. Connect the camera to the host PC with a USB cable and display the terminal
2. Execute the lsusb command with the v option to see information about the various USB devices.

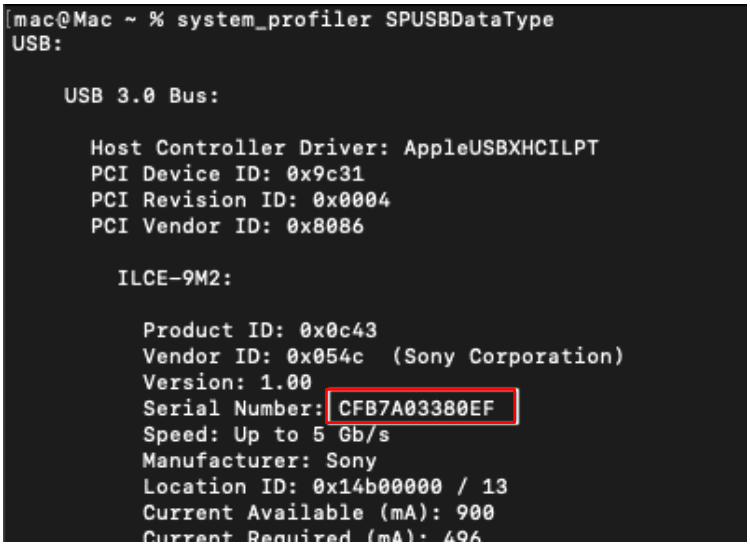
Area marked in red : iSerial




**- macOS**

1. Connect the camera to the host PC with a USB cable and display the terminal
2. Execute the system_profiler command with the SPUSBDataType to see information about the various USB devices.

Area marked in red : Serial Number



*Example:*

```
void CreateUSBObject() {

        CrChar serialNum[(SCRSDK::USB_SERIAL_LENGTH + 1] = {0}; // +1 is Null-terminate

        memcpy(serialNum, L"CFB7A03380EF", sizeof(serialNum)); // wide char on Windows

        SCRSDK::ICrCameraObjectInfo* pCam = nullptr;

        CrError err = SCRSDK::CreateCameraObjectInfoUSBConnection(
                &pCam,
                SCRSDK::CrCameraDeviceModel_ILCE_9M2,
                (CrInt8u*)serialNum);

        if (CrError_None == err && pCam != NULL) {

                // connect to camera



                :
```

CreateCameraObjectInfoEthernetConnection() has five parameters. The second parameter specifies the model type of the camera to connect to, the third parameter specifies the IP Address of the camera to connect to, and the fourth parameter specifies the MAC address. Check the MAC address with the camera. Fifth parameter specifies the SSH authentication enable flag. The camera object pointer created by these conditions is returned with the first parameter ICrCameraObjectInfo **.

You can check the MAC address of the camera by the following method.

MENU > Network > Wired LAN > Display Wired LAN Info.

For ILME-FX6, it is the MAC address of the Wi-Fi adapter
MENU > Network > Wireless LAN > MAC Address

*Example:*

```
void CreateEthernetObject() {

        CrInt32 ipAddr = 0x0500A8C0; // 192.168.0.5

        CrInt8u macAddr[6] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06};

        SCRSDK::ICrCameraObjectInfo* pCam = nullptr;

        CrError err = SCRSDK::CreateCameraObjectInfoEthernetConnection(
                &pCam,
                SCRSDK::CrCameraDeviceModel_ILCE_9M2,
                ipAddr,
                macAddr,
                SCRSDK::CrSSHsupport_OFF);

        if (CrError_None == err && pCam != NULL) {

                // connect to camera



                :
```

If a “ICrCameraObjectInfo” created with incorrect information is used in Connect(), the SDK operation is not guaranteed.

# Connect a Camera

Using one of the enumerated ICrCameraObjectInfo, the camera can be connected with Camera Remote SDK by calling the Connect() function of the class. This function has five parameters. The first parameter ICrCameraObjectInfo * specifies the camera to connect to. The second parameter IDeviceCallback is a function object that is called back to notify the communication events from Camera Remote SDK. The caller must create the object instance before calling the Connect() function. The third parameter CrDeviceHandle * is returned with the connection handle from SDK and it must be set NULL before calling the Connect() function. The fourth parameter CrSdkControlMode is optional. To control the camera remotely, do not specify this parameter, or specify Remote Control Mode. Specify ContentsTransferMode to pull the content on the media inserted in the slot of the camera. The fifth parameter CrReconnectingSet is optional. You can specify whether to automatically reconnect after the connection with the camera is unintentionally lost. If not specified, the default is CrReconnecting_ON and automatic reconnection is performed. However, CrReconnecting_ON is valid only in RemoteControlMode. In the ContentsTransferMode, automatic reconnection is not performed regardless of the fifth parameter setting. The 6th to 9th parameters are all for SSH authentication. These parameters are not needed for cameras that do not require SSH authentication. Check "Supporting physical layer" to see if SSH authentication is required.

After the Connect() function, ICrCameraObjectInfo can be freed. There is no need to wait for OnConnected() or the OnError() callback function. It means you can delete the ICrEnumCameraObjectInfo object returned from the EnumCameraObjects() function.

*Example:*

```
class MyDeviceCallback : public IDeviceCallback {
        void OnConnected(DeviceConnectionVersion version) {
                DeviceConnectionVersion ver = version;
                // Program can use the device handle.
            };
        :
bool ConnectCamera(ICrCameraObjectInfo *pcamera) {
        MyDeviceCallback *cb = new MyDeviceCallback();
        CrDeviceHandle          hDev = NULL;
        CrError err = SCRSDK::Connect(pcamera, cb, &hDev);
```

The following is an example of a ContentsTransferMode connection.

*Example:*

```
CrError err = SCRSDK::Connect(pcamera,cb,&hDev,CrSdkControlMode_ContentsTransfer);
```

Switching between RemoteControlMode, ContentsTransferMode, and CrReconnectingSet cannot be performed while connected. After disconnecting in each mode, reconnect in the desired mode.

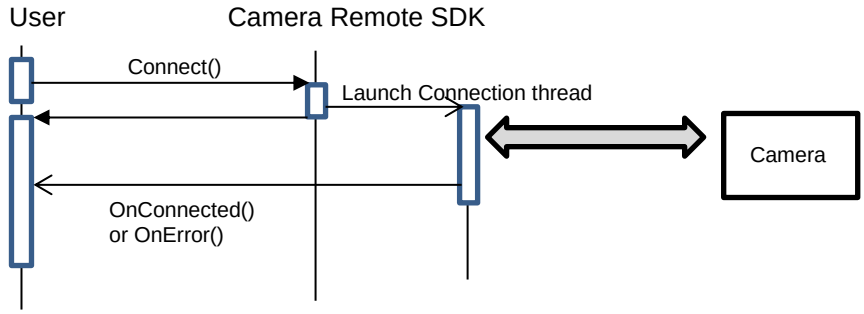
The following is an example of connecting to an SSH certified camera.

For SSH authentication, you need to get the data for the 8th parameter and 9th parameter of Connect() in advance with GetFingerprint(). The user needs to check that the fingerprint data obtained from the camera is correct. If fingerprint data different from the fingerprint data owned by the camera is returned by the GetFingerprint(), please do not proceed to Connect().
If you do not check whether the fingerprint data obtained by Get Fingerprint() is legitimate and specify incorrect fingerprint data in Connect(), the security of the host PC is not guaranteed.
For the 6th parameter userId and the 7th parameter userPassword, use the information set in the camera body.

*Example:*

```
CrInt32u fpLen = 0;
char fpBuff[128] = { 0 };
CrError err = SCRSDK::GetFingerprint(pcamera, fpBuff, &fpLen);
if (CR_SUCCEEDED(err)) {
    // If the acquired Fingerprint is correct, proceed to connection
    . . .

    err = SCRSDK::Connect(pcamera,cb,&hDev,CrSdkControlMode_Remote,CrReconnecting_ON,
            “admin”, “12345678”, fpBuff, fpLen);
            . . .
```

As described at the top of this section, the connection process is executed asynchronously. Calling the Connect() function means that just the connection process is started. When the connection is established, the OnConnected() callback of IDeviceCallback is called.




The left vertical line indicates the user thread of your program, the center vertical line indicates API of Camera Remote SDK, and the right vertical line indicates the camera connection thread inside Camera Remote SDK.

Connect() returns an error when the function parameter is not valid. In the synchronous process in the Connect() function, it does not check for the device existence or the connectivity. It is checked in the Connection thread. If the camera is not found or if the camera is not compatible with the Camera Remote SDK, the OnError() callback function is called with an error id, CrError_Connect_Connect.

If the connection is established, the OnConnected() callback function is called with a parameter for connecting Remote Control Protocol Version.

In this Camera Remote SDK version, the parameter's value below is fixed.

*Device_Connection_Version_RCP3 = 300*

Because this version's Camera Remote SDK supports only the Remote-Control Protocol Version 3.

The camera may not accept shooting operations immediately connection.

# Disconnect a Camera

Call the Disconnect() function to disconnect the camera. The function needs one parameter for the DeviceHandle to disconnect.

*Example:*

```
void Disconnect(CrDeviceHandle handle) {
        SCRSDK::Disconnect(handle);
```

If the handle is not valid, Disconnect() returns an error.

Disconnect() is also an asynchronous process. The return from Disconnect() does not mean that the camera has been disconnected. At the time of the OnDisconnected() callback function is called, the camera has been disconnected from the Camera Remote SDK.

See the table on the next page for the connection status of the camera and Camera Remote SDK.

## Changes in Camera Remote SDK connection status

The table below shows the connection status of the Camera Remote SDK, using some cases of connection and disconnection between the Camera Remote SDK and the camera as examples.

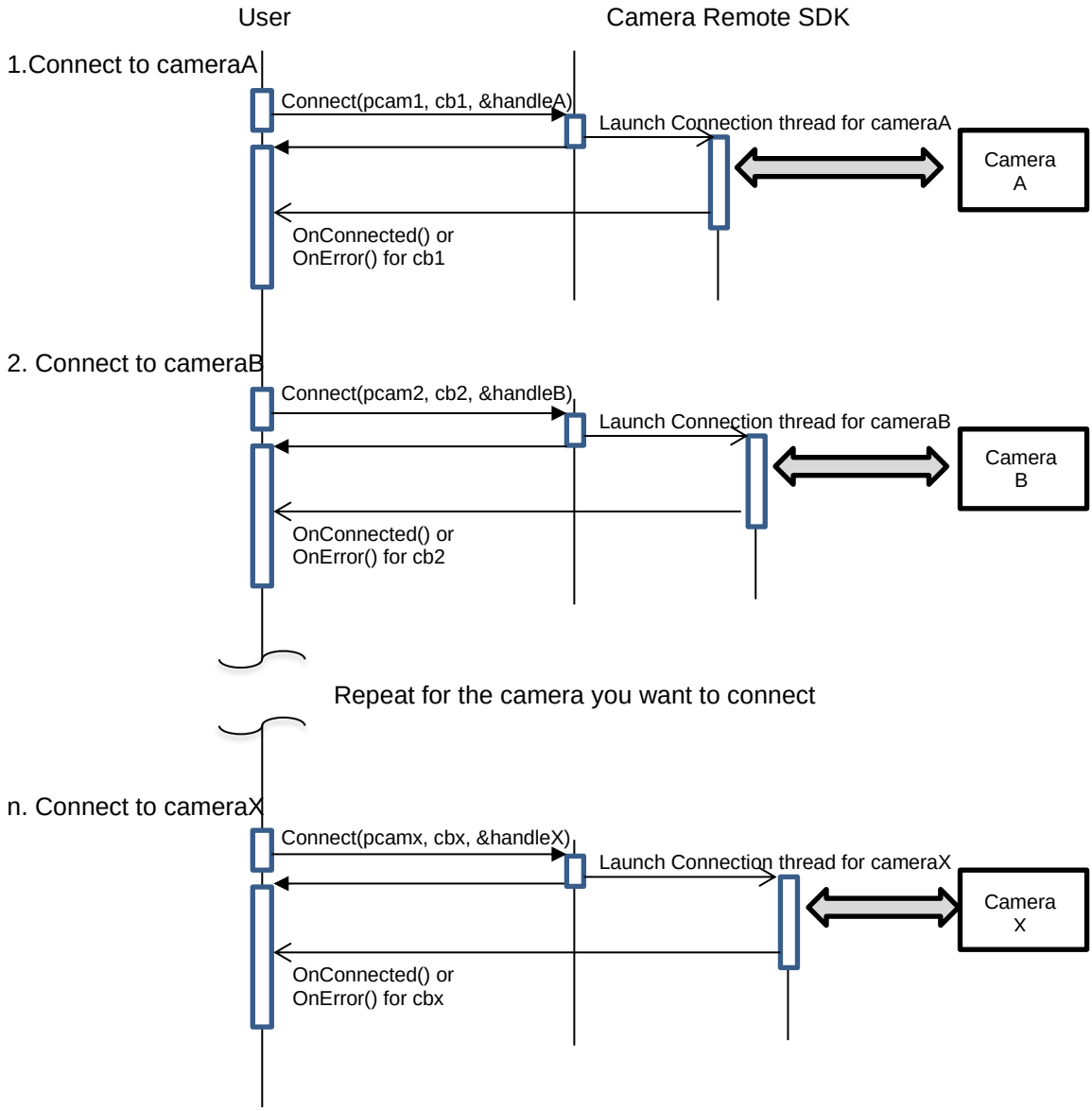
| No. | User operation | Physical (USB) | Camera Remote SDK | | | | |
|---|---|---|---|---|---|---|---|
| | | | Connection status with the camera | DeviceHandle | Camera communication thread | | |
| | | | | | | Main Loop (*1) | Sub Loop (*2) |
| Case 1 | Connect/Disconnect transition | | | | | | |
| 1 | Connect the camera to the PC | **Disconnected -> Connected** | - | - | - | | |
| 2 | Call Connect() function | **Connected** | **Disconnected -> Connected** | ✓ (generate) | **run** | run | stop |
| 3 | Call Disconnect() function | **Connected** | **Connected -> Disconnected** | ✓ | **stop** | stop | stop |
| 4 | Call ReleaseDevice() function | **Connected** | - | - (removed) | - (removed) | | |
| Case 2 | Physical disconnect and recovery transition | | | | | | |
| 1 | Connect the camera to the PC | **Disconnected -> Connected** | - | - | - | | |
| 2 | Call Connect() function | **Connected** | **Disconnected -> Connected** | ✓ (generate) | **run** | run | stop |
| 3 | Remove the USB cable | **Connected -> Disconnected** | **Connected -> Reconnecting** | ✓ | **run** | stop | run |
| 4 | Reconnect the USB cable | **Disconnected -> Connected** | **Reconnecting -> Connected** | ✓ | **run** | run | stop |
| Case 3 | Physical disconnect and timeout transition | | | | | | |
| 1 | Connect the camera to the PC | **Disconnected -> Connected** | - | - | - | | |
| 2 | Call Connect() function | **Connected** | **Disconnected -> Connected** | ✓ (generate) | **run** | run | stop |
| 3 | Remove the USB cable | **Connected -> Disconnected** | **Connected -> Reconnecting** | ✓ | **run** | stop | run |
| 4 | 5 minutes passed | **Disconnected** | **Reconnecting -> Disconnected** | ✓ | **stop** | stop | stop |

*1 : Data transmission / reception such as acquiring and updating Device Property and acquiring LiveView Image.
*2 : Monitoring reconnection. This is valid in “Remote Control Mode”. “Content Transfer Mode” does not monitor reconnection.

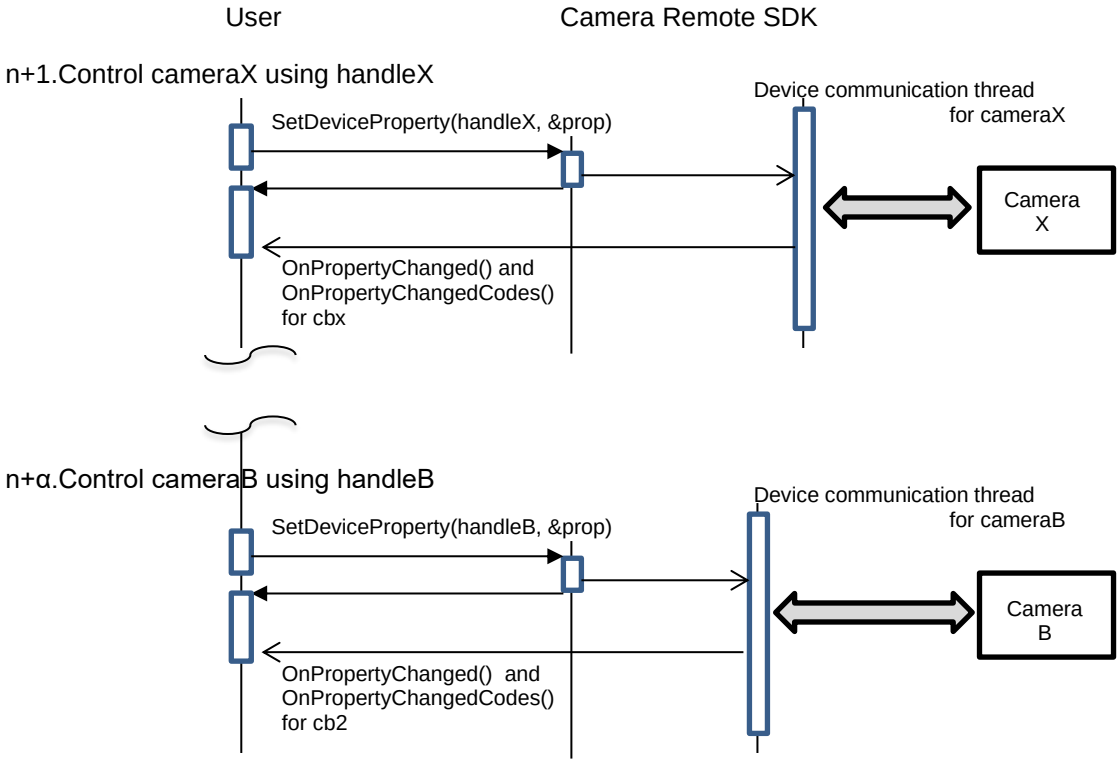
Note : If CrReconnecting_OFF is specified for the fourth parameter of the Connect(), automatic reconnection will not be performed in all cases.

## Connect/Disconnect multiple cameras

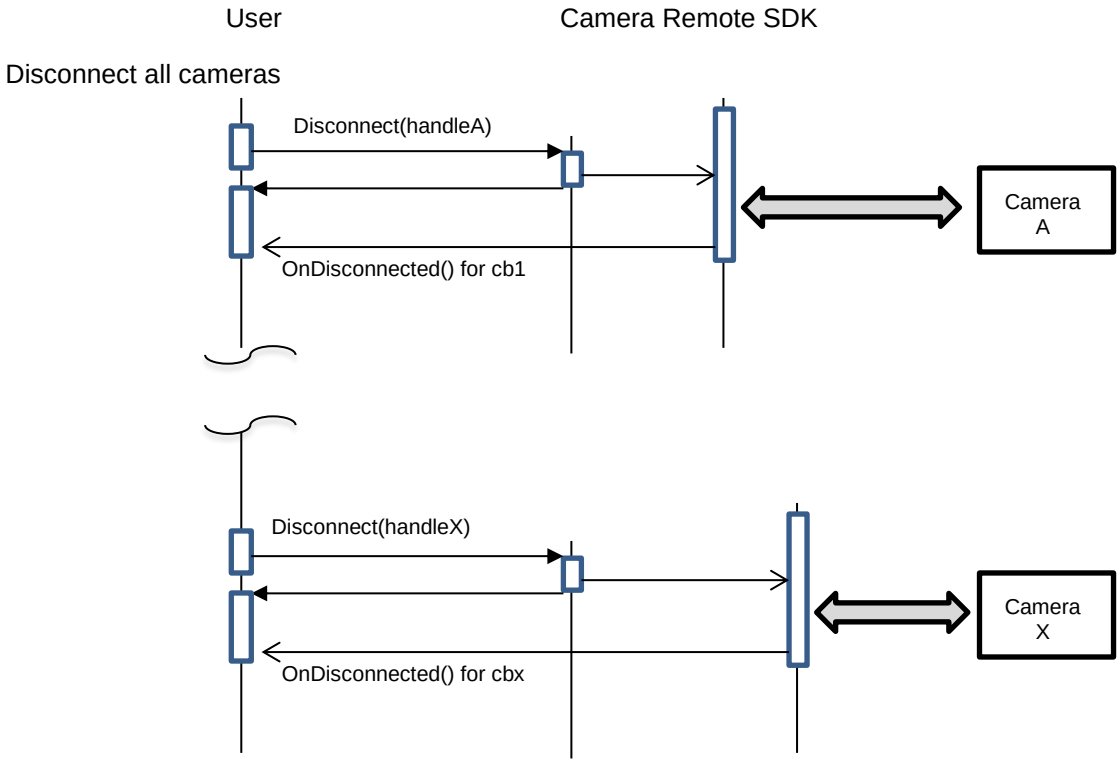
To control multiple cameras, call the Connect() function for the number of cameras and get a handle for the number of cameras.
With each handle you get, you can control each camera.




After that, use handleA to handleX to control cameraA to cameraX




When ending control of multiple cameras, use all handles to call the Disconnect() function to disconnect from all cameras.



Points to note when connecting USB

- Pay attention to the maximum power supply of the USB bus controller and the power consumption required by the camera

- When using multiple cameras at the same time, it is recommended to turn off the USB power supply setting on the cameras.
  “USB Power Supply” menu varies by model. Please refer to the help guide for your model.

- Multiple connection requires more CPU loads than single connection, and there is a possibility to cause delays in getting and updating properties. If you do not need to display LiveViewImage from all cameras at the same time, it is recommended to disable LiveViewImage acquisition to reduce the processing load by using SetDeviceSetting.
  refs SDK Properties

# Get the Camera Properties

After the connection is established, camera property can be acquired by the GetDeviceProperties() function. This function has three parameters. The first one is the device handle that specifies the device, the second one is the pointer to CrDeviceProperty pointer that receives the acquired property list, and the third one receives the size of the CrDeviceProperty list.

The CrDeviceProperty returned from GetDeviceProperties() is allocated in Camera Remote SDK and the memory MUST be freed by calling ReleaseDeviceProperties() function.

*Example:*

```
void GetProperties(CrDeviceHandle handle) {
    CrDeviceProperty *pProperties;
    CrInt32u numofProperties = 0;
    SCRSDK::GetDeviceProperties(handle, &pProperties, &numofProperties);
    if (pProperties) { // the property list is received successfully
      for (CrInt32u n = 0; n < numofProperties; n++) {
        SCRSDK::CrDataType type = pProperties[n].GetValueType();
        int dataLen = sizeof(CrInt64u); // Maximum length
        if (type & SCRSDK::CrDataType_UInt8) {
          dataLen = sizeof(CrInt8u);
        } else if (type & SCRSDK::CrDataType_UInt16) {
          dataLen = sizeof(CrInt16u);
        :
        }
        int numofValue = pProperties[n].GetValueSize() / dataLen;
        switch (pProperties[n].GetCode()) {
        case CrDeviceProperty_FNumber:
        // code to parse the properties...
            :
    }
    SCRSDK::ReleaseDeviceProperties(handle, pProperties);
```

In the sample code above, for code simplification, the return value of GetDeviceProperties() is not checked, but it has to be checked. If the camera has already disconnected, it returns CrError_Invalid_Parameter. Additionally, in case of device property memory allocation error, it returns CrError_Generic_Unknown.

The content of the property list depends on the camera features. It is not expected that all of the properties are defined in enum of CrDevicePropertyCode in CrDeviceProperty.h. Some properties defined in CrDevicePropertyCode will also be acquired by the GetLiveViewProperties() function as described in the following section.

This function does not communicate with the camera. This function returns the copy of the latest property list. The camera properties are updated automatically inside this Camera Remote SDK. In case of one or other properties are changed, Camera Remote SDK calls OnPropertyChanged() and more callback functions. Camera Remote SDK assumes that GetDeviceProperties() is called at the beginning of the camera operation, and when Camera Remote SDK calls the OnPropertyChanged() call back function. But calling the GetDeviceProperties() function in the OnPropertyChanged() or other callback function is not recommended, because the callback function is called on the thread that communicates with the camera. All callback functions are expected to return as soon as possible.

The following sample code is one of the references to get updated properties and to update the user interface items in Windows.

```
Example:
void MyDeviceCallback::OnConnected() {
        ::PostMessage(wnd, WM_APP_UPDATE_PROPERTIES, 0L, 0L);
}
void MyDeviceCallback::OnPropertyChanged() {
        ::PostMessage(wnd, WM_APP_UPDATE_PROPERTIES, 0L, 0L);
}

ON_MESSAGE(WM_APP_UPDATE_PROPERTIES, OnMessageUpdateProperties)

void CAppWnd::OnMessageUpdateProperties(WPARAM wp, LPARAM lp)
{
        CrDeviceProperty *pProps;
        CrInt32u numofProps = 0;
        GetDeviceProperties(handle, &pProps, &numOfProps);
        :  // update user interface items
```

The following sample code is an example using the API and callback functions added from Version 1.05.00.
It is possible to call the GetSelectDeviceProperties() with the information notified in the OnPropertyChangedCodes callback to get only the specified device properties.

```
Example:
void MyDeviceCallback:: OnPropertyChangedCodes(CrInt32u num, CrInt32u* codes) {
        COPYDATASTRUCT cds;
        cds.dwData = 0;
        cds.lpData = codes;
        cds.cbData = num * sizeof(CrInt32u);
        ::SendMessage(wnd, WM_APP_UPDATE_PROPERTIES_DIFF, num, &cds);
}

ON_MESSAGE(WM_APP_UPDATE_PROPERTIES_DIFF, OnMessageUpdateProperties2)

void CAppWnd::OnMessageUpdateProperties2(WPARAM wp, LPARAM lp)
{
        CrDeviceProperty *pProps;
        CrInt32u numofProps = 0;
        GetSelectDeviceProperties(handle, wp, lp, &pProps, &numofProps);
        :  // update user interface items
```

# Get the Live View Properties

Some camera properties cannot be acquired by the GetDeviceProperties() function. The properties that are defined in CrLiveViewPropertyCode are independent from the device property list, and must use the GetLiveViewProperties() function, because those properties are strongly related to the live view image.

The function interface and the usage are similar to GetDeviceProperties().

Similar to the device properties, the memory object returned from GetLiveViewProperties() must also be freed by ReleaseLiveViewProperties().

*Example:*

```
void GetLiveViewProperties(CrDeviceHandle handle) {
        CrLiveViewProperty *pProperties = NULL;
        CrInt32u numofProperties = 0;
        SCRSDK::GetLiveViewProperties(handle, &pProperties, &numofProperties);
        if (pProperties) {// the property list is received successfully
                for (CrInt32u n = 0; n < numofProperties; n++) {
                        switch (pProperties[n].GetCode()) {
                        case CrLiveViewProperty_AF_Area_Position:
                        // code to parse the properties...
        :
                }
                SCRSDK::ReleaseLiveViewProperties(handle, pProperties);
        }
```

# Device Properties and Live View Properties

CrDeviceProperty class and CrLiveViewProperty class store similar property values. The contents and the differences are explained in this section.

The CrDeviceProperty class has the following member variables shown below:

- code : Identify the content of the property.
- value Type : Specify the value variable type.
- enable Flag : Capability of the operation. Modifiable / Get Only / Invalid / Set Only
- current Value : Current property value. This value is defined as a 64bit variable.

If the property has a limited number of options, it has a list of the selectable options.

- value Size : Number of the selectable options.
- values : List of the selectable options.

The property code is defined in enum CrDevicePropertyCode in CrDeviceProperty.h. For example, CrDeviceProperty_FNumber is defined as 0x0100. The value type is CrDataType_UInt16. The current value is defined as a 64bit variable, but in this case only the highest 16bit is valid.

*Example:*

```
switch (property->code) {
case CrDeviceProperty_FNumber:
    CrInt16u currentvalue = static_cast<CrInt16u>(property->GetCurrentValue());
    :
```

If the enable flag is modifiable, the property can be acquired and can be set. To change the property value, refer to the SetDeviceProperty() function described in the next section. If the enable flag is Get Only, the property can be acquired and be referred to by GetDeviceProperties(), but cannot be changed.

Invalid means the property is invalid. This property must not be referred to or set. Set Only is also a very special case, as you see there is no “SetLiveViewProperty()” function. The properties you get via GetLiveViewProperties() are properties closely related to the live view feature, but in order to change the property you can use the SetDeviceProperty() function.

Depending on the camera status, this flag value changes. In case of CrDeviceProperty_FNumber, if the exposure mode of the camera is “M” or “A”, this flag is modifiable, and in case of “P” or “S”, this flag is Get Only.

If the property has selectable options, it has the list and the count of the list. Please note that the size is “Byte Size”, not the count of the elements. Therefore, dividing by the size of the value type, the count of the elements can be calculated.

See the following reference pages to understand the property code and the type definitions.

*Example:*

```
switch (property->code) {
case CrDeviceProperty_FNumber:
        CrInt16u currentvalue = static_cast<CrInt16u>(property->GetCurrentValue());
        CrInt32u countofelement = property->GetValueSize() / sizeof(CrInt16u);
        CrInt16u *poptions = static_cast<CrInt16u*>property->GetValues();
        if (countofelement) {
                CrInt16u *elements = new CrInt16u[countofelement];
                for (CrInt32u n = 0; n < countofelement; n++) {
                        elements[n] = *poptions++;
```

The CrLiveViewProperty class has similar members but there is “value size” to specify the memory size of current value, and there is no “selectable option” and its size field.

- code : Identify the content of the property.
- value Type : Specify the frame data type of value.
- enable Flag : Capability of the operation. Get Only
- value Size : Memory size in Bytes of Current property value.
- value : Current property value. This value is a memory block.

This value size is larger than CrDeviceProperty, because CrLiveViewProperty has the properties that represent coordination, regions or in some cases include the style. The definitions of the data type are described in the header file of “CrDeviceProperty.h” and the following reference section.

Because this CrLiveViewProperty class tells the information of the focus area, live view display magnification region, or custom white balance region, the API to get the properties from the camera is separated from GetDeviceProperties().

But note that to change those properties, the SetDeviceProperty() command must be used.

*Example:*

```
switch (property->code) {
case CrLiveViewProperty_AF_Area_Position:
        CrFocusFrameInfo *pinfo
        = static_cast<CrFocusFrameInfo *>(property->value);
```

# Change the Camera Properties

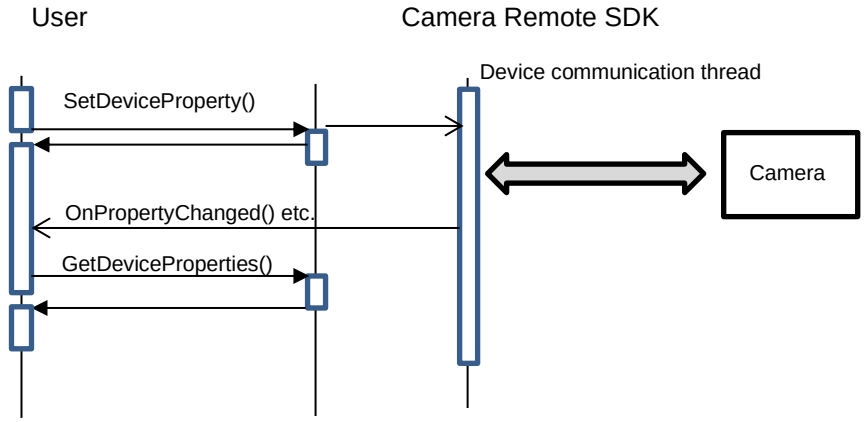
To change camera properties, for example F number, shutter speed, ISO and so on, send change property commands by using SetDeviceProperty(). There are two parameters, the first parameter is the device handle of the target camera, and the second parameter is the CrDeviceProperty class object. In this CrDeviceProperty object, only the code and value members are referred to in Camera Remote SDK.

If the value is invalid, the command is ignored, for example, where the out-of-range F number or setting F number in exposure mode is “S”.

The combination of the code and the available value is described in API Reference section.

Note that this SetDeviceProperty() call is not synchronous. Once SetDeviceProperty() is called, the command is queued in the command queue in Camera Remote SDK and it is transmitted to camera at the appropriate time. It means that there is a short time lag between this function call and the camera’s property change.

The properties in Camera Remote SDK are also not updated by the SetDeviceProperty() function. SDK keeps the property status of the camera. It is updated after the camera changes its status.




If the property is not changed because of the camera status, Camera Remote SDK does not notify you of anything. It is recommended to set the 3- to 5-second timer in the user interface and try to get the property status to SDK and update the user interface state.

The following is sample code for updating device property of numeric type.

*Example:*

```
SCRSDK::CrDeviceProperty prop;
prop.SetCode(code); // Specify the code of the device property you want to update
prop.SetValueType(type); // Specify the type of the device property you want to update
prop.SetCurrentValue((CrInt64u)newValue);
SCRSDK::SetDeviceProperty(handle, &prop);
```

The following is sample code for updating a device property of type CrDataType_STR.

*Example:*

```
        SCRSDK::CrDeviceProperty prop;
        prop.SetCode(code); // Specify the code of CrDataType_STR device property you want to update
        prop.SetValueType(type); // Specify the type of CrDataType_STR device property you want to update
#if defined (_UNICODE) || defined (UNICODE)
        std::wstring input(L“TEST1”);
#else
        std::string input(“TEST1”);
#endif
        // The string length that can be set varies depending on the device property.
        // Check the maximum character length of the device property to be updated.
        // The only CrDataType_STR device property that can be updated in version 1.08.00 is CrDeviceProperty_ImageID_String.
        int strLen = input.length();
        if (64 < strLen)
            return;  // String is too long.

        // Prepare a place to store the string to be set.
        // Notes:
        //      Append  a null-terminate and pass it to the Camera Remote SDK.
        //      The first 2 bytes are the character string length including the null-terminate.
        CrInt16u* setStr = new CrInt16u[strLen+2];
        memset(setStr, 0, sizeof(setStr));
        setStr[0] = (CrInt16u)strLen + 1; // +1 = null-terminate
#if defined (_UNICODE) || defined (UNICODE)
        lstrcpy((wchar_t*)&setStr[1], input.c_str());
#else
        // Convert multi byte char to wide byte char
        wchar_t wbuff;
        for (int i = 0; i < input.length(); ++i) {
            int retLen = mbtowc(&wbuff, &input.at(i), 1);
            if (-1 != retLen) {
                setStr[i + 1] = (CrInt16u)wbuff;
                //setStr[i + 1] = (CrInt16u)((wbuff & 0xFF) << 8 | ((wbuff >> 8) & 0xFF)); // For endian convert
            }
        }
#endif
        prop.SetCurrentStr(setStr); // Use SetCurrentStr() for CrDataType_STR
        SCRSDK::SetDeviceProperty(handle, &prop);
        delete[] setStr;
```

# Send a Control Command

Some of the camera commands are implemented as “Control Command”. For example, shutter release (fully pressing the shutter button), movie record and so on. In these cases, the SendCommand() function must be used. The interface of this function is much simpler than the device property case.

*void SendCommand(CrDeviceHandle device, CrInt32u commandId, CrCommandParam parm);*

The first parameter specifies the device, the second parameter indexes the command id and the last parameter is ON (CrCommandParam_Down) or OFF (CrCommandParam_Up). The Up and Down expresses the physical button action.

The following example shows how to capture images.

*Example:*

```
SCRSDK::SendCommand(handle, CrCommandId_Release, CrCommandParam_Down);
```

This command initiates a human’s action using the button; therefore, the button must be released (Up) once when you send “Down” command. If the camera’s drive mode is in the continuous shooting mode, the camera captures continuously what it receives from the CrCommandParam_Down command until it receives CrCommandParam_Up.

This sample code shows the simplest way to press the shutter release button for one second.

*Example:*

```
SCRSDK::SendCommand(handle, CrCommandId_Release, CrCommandParam_Down);
Sleep(1000);
SCRSDK::SendCommand(handle, CrCommandId_Release, CrCommandParam_Up);
```

This command sent by SendCommand() has a higher priority than other communication processes, getting device properties, and getting live view image data and so on, to make the response of camera quicker.

# Get a Live View Image

Live view image is sent from the camera as a Jpeg image. The image size depends on the live view image quality of the camera setting and the image aspect mode.

The image is updated at a rate of 30 frames per second if the communication speed is good. The FPS becomes much lower when the communication bandwidth is narrow. The situations, where the communication quality is poor or where captured images are transmitted, result in corresponding to a lower live view FPS.

To receive live view image, a receive buffer needs to be prepared. The buffer size can be acquired by the GetLiveViewImageInfo() function. The first parameter is the device handle, and the second parameter is the pointer to CrImageInfo. CrImageInfo has the information related to width, height and the required buffer size. After getting the image buffer size, allocate the memory buffer for the image and call GetLiveViewImage().

*Example:*

```
CrImageInfo *pInfo = new CrImageInfo();

SCRSDK::GetLiveViewImageInfo(handle, pInfo);

CrImageDataBlock *pLiveViewImage = new CrImageDataBlock();

pLiveViewImage->SetSize(pInfo->GetBufferSize());

CrInt8u* recvBuffer = new CrInt8u[pInfo->GetBufferSize()];

pLiveViewImage->SetData(recvBuffer);

SCRSDK::GetLiveViewImage(handle, pLiveViewImage);
```

*Example:*

```
SCRSDK::GetLiveViewImage(handle, pLiveViewImage);

CrInt32u size = pLiveViewImage->GetImageSize();

CrInt8u *pdata = pLiveViewImage->GetImageData();
```

CrImageInfo has the Jpeg image data and its size. GetImageData() returns the data pointer and GetImageSize() returns the data size.

This Jpeg image data starts from SOI marker (FF D8) and ends with EOI marker (FF D9). It can be displayed as it is by the graphic user interface using OpenGL, DirectDraw or another framework.

*Example:*

```
SCRSDK::GetLiveViewImage(handle, pLiveViewImage);
CrInt32u size = pLiveViewImage->GetImageSize();
CrInt8u *pJpegData = new CrInt8u[size];
memcpy(pJpegData, pLiveViewImage->GetImageData(), size);
```

The image is updated inside Camera Remote SDK and one unique and an incremental number is given for the image that is transmitted from the camera. GetLiveViewImage() compares the frame number of the given CrImageDataBlock class object and the current frame number in the Camera Remote SDK. If the given number is smaller than the current number, a copy of the new image buffer is made of the given object and updates the frame number of the given object. If the number of the object is equal or larger than the number of the SDK, no copy is made, and it returns CrWarning_Frame_NotUpdated. Therefore, at the first call of GetLiveViewImage(), the frame number of CrImageDataBlock should be set to zero.

The size member of CrImageDataBlock is updated to the real image data size in GetLiveViewImage(). Where the buffer size of CrImageDataBlock is smaller than received image size, Camera Remote SDK also does not copy the buffer and returns CrError_Memory_Insufficient.

If the return value of the GetLiveViewImage() is CrWarning_Frame_NotUpdated, wait for a while and get the frame again. If the return value is CrError_Memory_Insufficient, get the image buffer size by GetLiveViewImageInfo() and reallocate the memory as the new size.

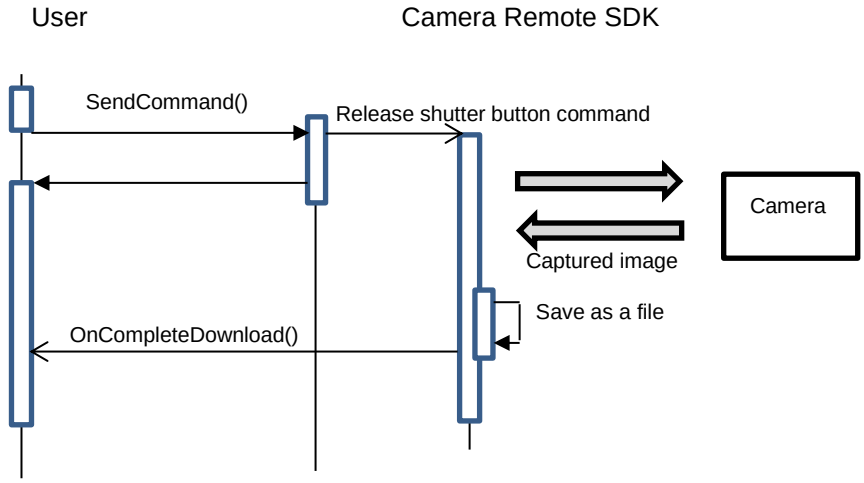
If GetLiveViewImage() returns CrError_Generic_Unknown, it means that there is an issue related to the data communication between the PC and Camera.

ILME-FX6 does not support GetLiveViewImage(), so LiveView(Video Monitoring) must be displayed via HDMI or SDI.

# Capture an Image Sequence

Where the store image destination (CrDeviceProperty_StillImageStoreDestination) property is “PC” or “PC and Camera”, the captured image is automatically transmitted to PC and stored in the storage of the PC by Camera Remote SDK.

This section explains the sequence of the storing captured images.




After Camera Remote SDK stored the image to a file, the OnCompleteDownload() callback function is called with the stored file path.

*void OnCompleteDownload(CrChar *filename);*

The store image folder can be set using the SetSaveInfo() function. The next section explains this process.

# Change the Store Image Folder and the File Name

Camera Remote SDK has two modes to specify the image file name. One is “Auto Mode” and the other is “Manual Mode”.

Auto Mode gives the image file name that is determined by the camera. In this case the naming rule of the camera is used. If the file name conflicts with an existing file, an additional number is appended after the file name like DSC01234(1).JPG.

In Manual Mode, your program can specify the file name prefix and the start number. “ABCDE” as prefix and 100 as the start number makes the name from “ABCDE00100.JPG”. To change the mode and the prefix and start number, use the SetSaveInfo() function. In this case, Camera Remote SDK finds a number that does not conflict with existing files and incrementally sets the file number like ABCDE00100(1).JPG.

The SetSaveInfo() function has four parameters. The first parameter specifies the device handle, the second parameter specifies the folder path to store, the third parameter specifies the file prefix string and the last parameter specifies the start number that is added to the file name.

To change to Auto Mode, set the null string (note that it means “”, not null pointer) and give -1 as the start number.

*Example:*

```
SCRSDK::SetSaveInfo(handle, L“C:\Image”, L””, -1);
```

Using Manual Mode and the specified prefix, set the string of the parameter. For example, to store the images in “C:\Image”, set the string giving the “ABCDE” prefix and the sequential number from 00100.

*Example:*

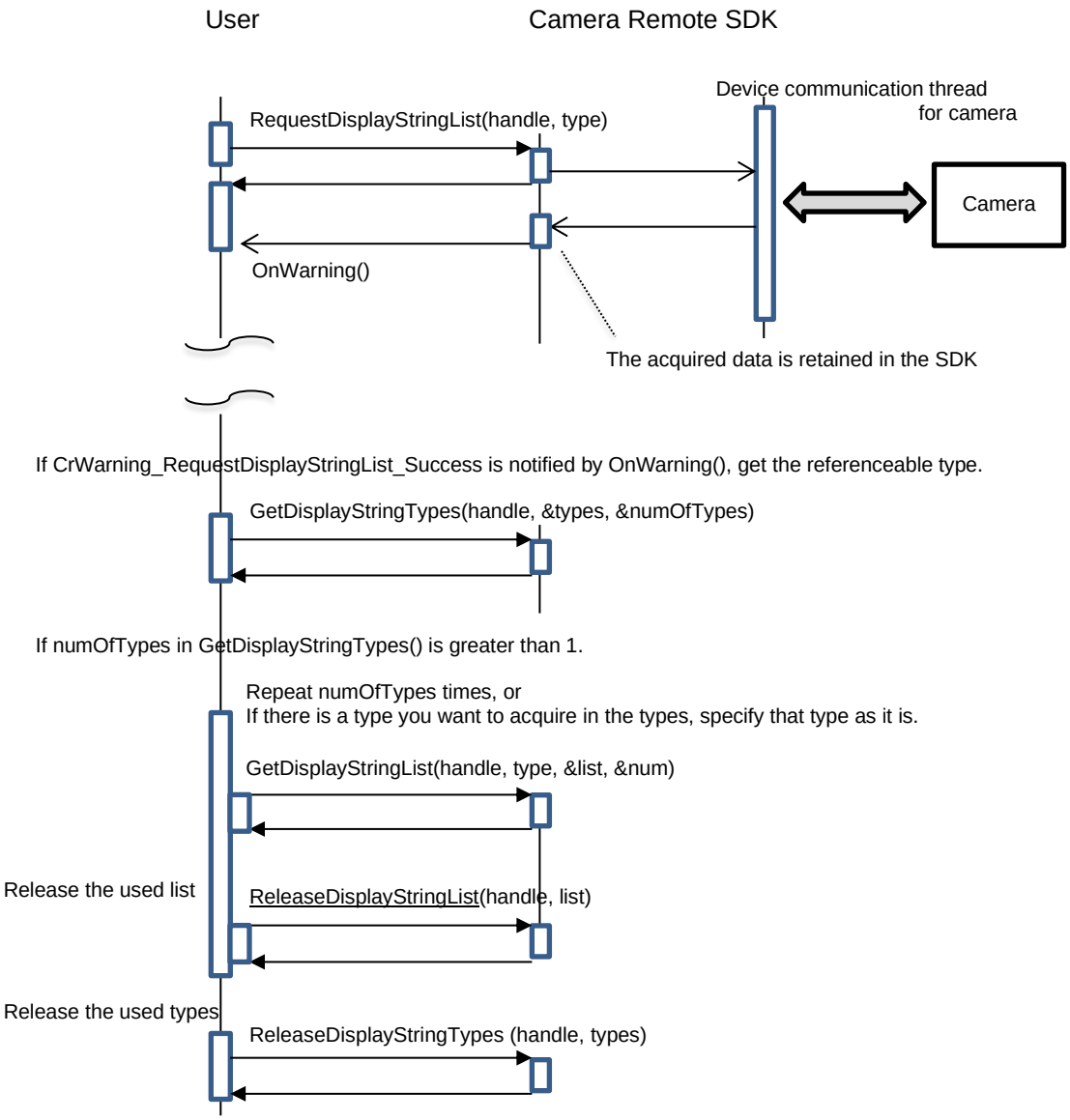
```
SCRSDK::SetSaveInfo(handle, L“C:\Image”, L”ABCDE”, 100);
```

Camera Remote SDK works in Unicode, the folder path and the prefix must be set as Unicode string.

# Get the menu display string

Information (character string and value) related to the menu display of the camera body can be acquired. It is assumed that the acquired information will be used in each user application.

First, request the SDK to get the display string information from the camera body with RequestDisplayStringList(). The result will be notified by a OnWarning().
If the request is successful, you will be able to know the types of information that can be obtained with GetDisplayStringTypes(), and you will be able to get the information with GetDisplayStringList(). It is recommended to check the types that can be referenced by GetDisplayStringTypes() before doing GetDisplayStringList().



*Example:*

```
std::map<int, std::string> m_baseIsoList;

CrError err = SCRSDK::RequestDisplayStringList(
                handle,
                SCRSDK:: CrDisplayStringType_Camera_Gain_BaseISO_Display);
```

*When the OnWarning callback notifies you of success:*

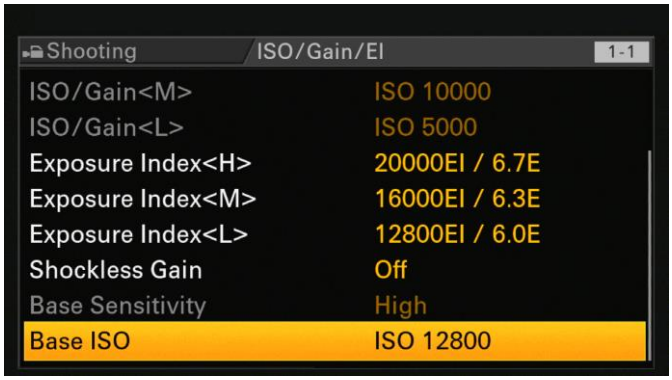
```
CrInt32u numOfTypes = 0;

SCRSDK::CrDisplayStringType* types = nullptr;

CrError err = SCRSDK::GetDisplayStringTypes(
                handle,
                &types,
                &numOfTypes);

if (CR_SUCCEEDED(err) && 0 < numOfTypes) {

   CrInt32u numOfList = 0;

   CrDisplayStringListInfo * list = nullptr;

   err = SCRSDK::GetDisplayStringList(
                handle,
                types[0],
                &list,
                &numOfList);

   if (CR_SUCCEEDED(err) && 0 < numOfList) {

      // update menu variable   etc.

      std::string str((char*)list[i].displayString);

      m_baseIsoList.insert(std::pair<int, std::string>(
                (int)list[i].value, str));

....

      // release of list pointer

      SCRSDK::ReleaseDisplayStringList(handle, list);

    }

   // release of types pointer

   SCRSDK:: ReleaseDisplayStringTypes(handle, types);

}
```

The Gain BaseISO name obtained by GetDisplayStringList corresponds to the string displayed in the menu.

*Example:*

```
for (auto item = m_baseIsoList.begin(); item != m_baseIsoList.end(); ++item) {
        printf("[%d] %s\n", item->first, item->second);
}
```

ex. ILME-FX6 Gain BaseISO Menu



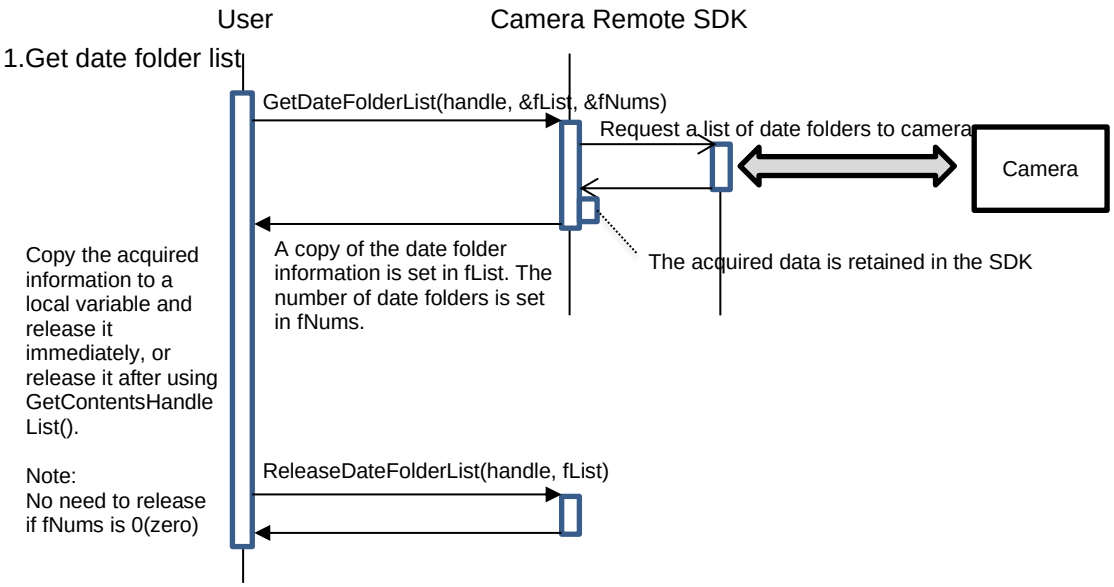

# Pull out content stored on media

When you connect in ContentsTransferMode, you can pull content from the media inserted in the camera slot.

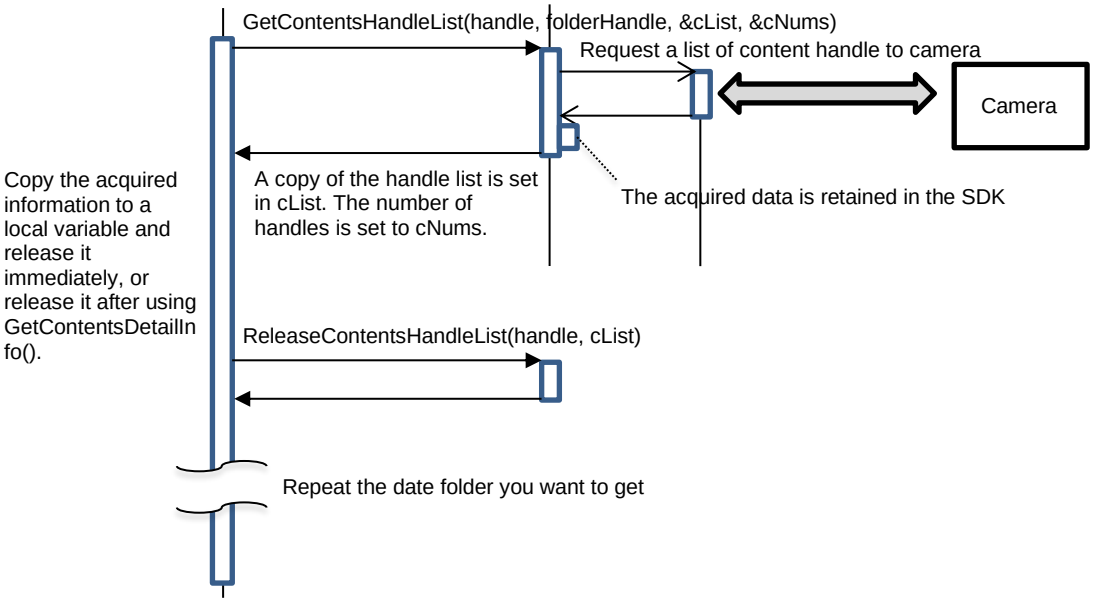
In order to pull the contents from the media, a content handle is required to identify the content.

Content/content handle is managed for each DateFolder. First, get the DateFolder list with GetDateFolderList(), and then use the DateFolder handle to get the handle list of the contents existing in the DateFolder with GetContentsHandleList().

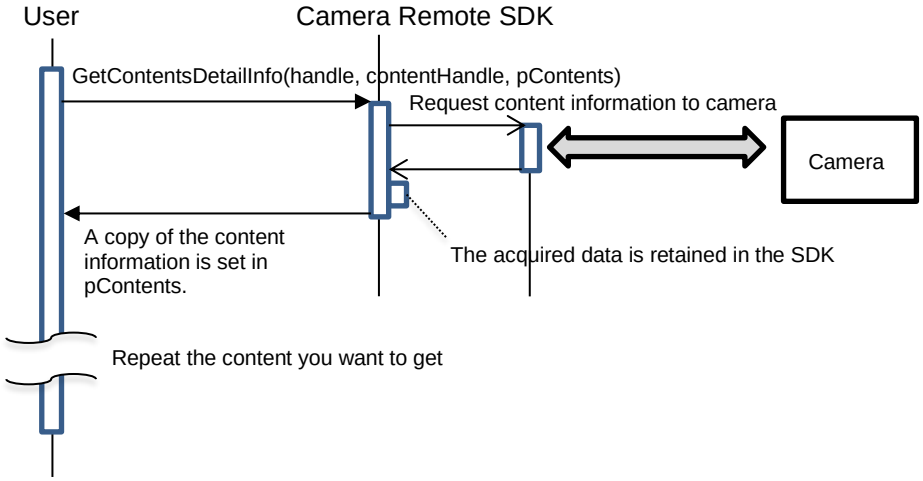
To know the file name and size of the content, get the detailed information with GetContentsDetailInfo().




2.Get a handle list of content that exists in the date folder



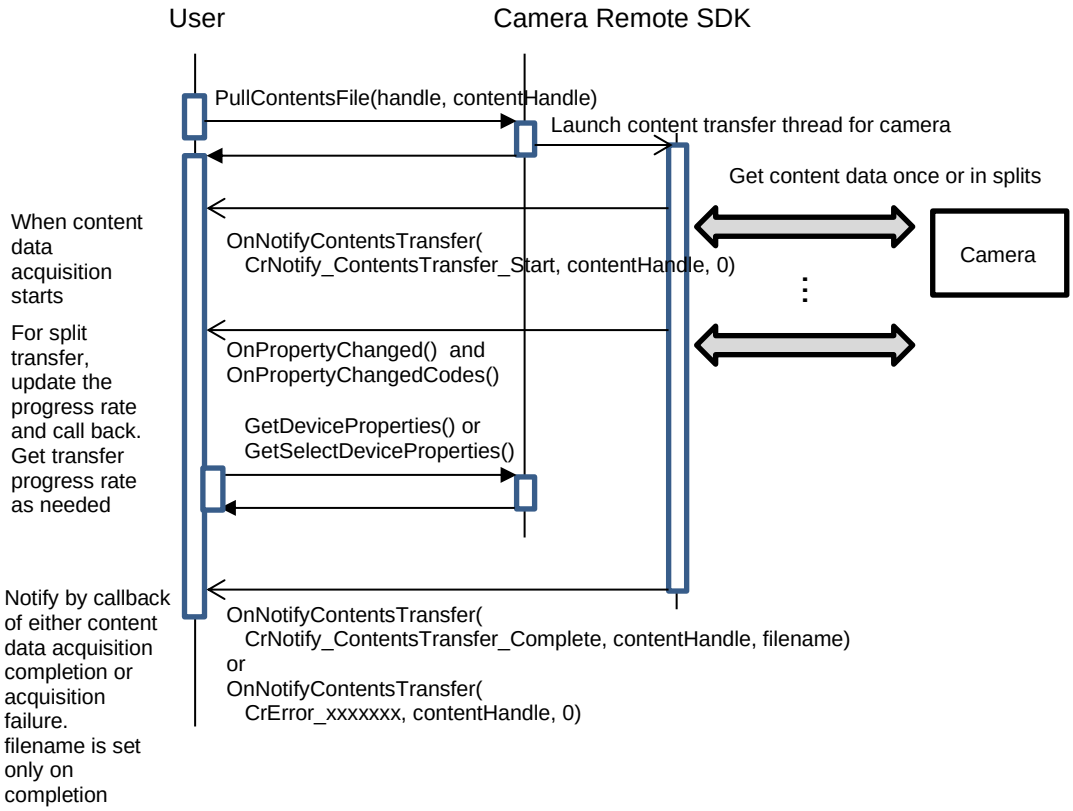
3. Get content detail information




*Example:*

```
CrInt32u fNums = 0;
SCRSDK::CrMtpFolderInfo* fList;
CrError err = SCRSDK::GetDateFolderList(handle, &fList, &fNums);
if (CR_SUCCEEDED(err)) {
    for (int i = 0; i < fNums; ++i) {
        CrInt32u cNums = 0;
        CrContentHandle* cList;
        err = SCRSDK::GetContentsHandleList(handle, fList[i].handle, &cList, &cNums);
        if (CR_SUCCEEDED(err)) {
            for (int j = 0; j < cNums; ++j) {
                SCRSDK::CrMtpContentsInfo* pContents = new SDK::CrMtpContentsInfo();
                err = SCRSDK::GetContentsDetailInfo(handle, cList[j], pContents);
                if (CR_SUCCEEDED(err))
                    m_contentList.push_back(pContents);
            }
            SCRSDK::ReleaseContentsHandleList(handle, cList);
        }
    }
    SCRSDK::ReleaseDateFolderList(handle, fList);
}
```

Save the content file to the host device using PullContentsFile(). PullContentsFile() is asynchronous. When the transfer is complete, you will be notified with the OnNotifyContentsTransfer() callback. When the user requests to cancel the content transfer or the connection is lost, the OnNotifyContentsTransfer() callback will notify you of the reason why it could not be completed.




Note: We cannot guarantee the transfer of content taken with other cameras. And large files may not be handled depending on the OS.

*Example:*

```
err = SCRSDK::PullContentsFile(handle, cList[j]);



class MyDeviceCallback : public IDeviceCallback {
    void OnNotifyContentsTransfer(CrInt32u notify, CrContentHandle handle, CrChar* filename = 0) {
        if (CrNotify_ContentsTransfer_Start == notify) {
            :
        else if (CrNotify_ContentsTransfer_Complete == notify) {
            :
        else
            : // Failure
```

You can also get a thumbnail of the content with GetContentsThumbnailImage(). For example, as a means of selecting the content to be pull, it is possible to preview the thumbnails of all the content on the application screen.

*Example:*

```
CrInt32u bufSize = 0x4B000; // Uses LiveViewImage buffer size
auto* image_data = new SCRSDK::CrImageDataBlock();
if (image_data) {
    CrInt8u* image_buff = new CrInt8u[bufSize];
    if (image_buff) {
        image_data->SetSize(bufSize);
        image_data->SetData(image_buff);
        SCRSDK::GetContentsThumbnailImage(handle, cList[j], image_data);
 :
```

Note that PullContentsFile() is an asynchronous API and GetContentsThumbnailImage() is a synchronous API. Camera Remote SDK will not be able to respond to GetContentsThumbnailImage() calls until it has completed the queue processing accumulated by one or more PullContentsFile() calls. And while running GetContentsThumbnailImage(), the application cannot call PullContentsFile().

# Get the MediaProfile

It is an API to get the MediaProfile stored in the media of the camera.

In ILME-FX6, meta information such as recorded content is called "MediaProfile".

With this API you can only get MediaProfile about the content. Not an API to get content files.

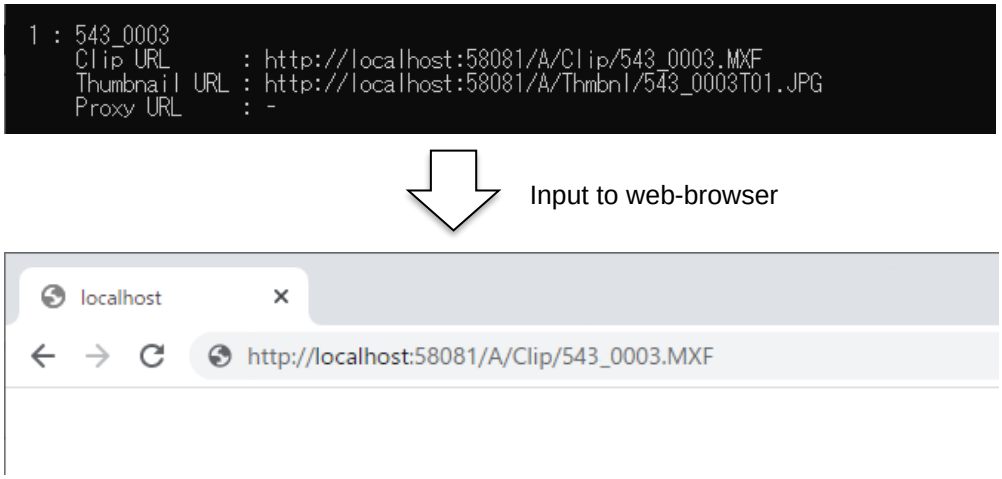
The second parameter specifies the Slot for which you want to get the MediaProfile. The third parameter is a pointer to which the list information of the acquired MediaProfile is written. The fourth parameter is set to the number of acquired MediaProfile and returns.

*Example:*

```
CrInt32u numOfList= 0;
SCRSDK:: CrMediaProfileInfo* mediaList;

CrError err = SCRSDK:: GetMediaProfile (
                    handle,
                    SCRSDK::CrMediaProfile_Slot1,
                    &mediaList,
                    &numOfList);

if (CR_SUCCEEDED(err) && 0 < numOfList) {

      //  etc.

      // release of list pointer

      SCRSDK:: ReleaseMediaProfile(handle, mediaList);

}
```

For example, there is "contentUrl" in the information obtained by this API. If you enter those URLs in browser software (Chrome, Safari, etc.) that supports streaming playback, you can play the content.

# Below is an example of Sample Application output



# SDK Properties

Using SetDeviceSetting(), some behavior of Camera Remote SDK can be changed. The setting can be set for each device.

*CrError SetDeviceSetting(CrDeviceHandle handle, CrInt32u key, CrInt32u value);*

The following code sample disables and enables the Live View function; setting “CrDeviceSetting_Disable” disables it and setting “CrDeviceSetting_Enable” enables it.

*Example:*

```
SCRSDK::SetDeviceSetting(handle, Setting_Key_EnableLiveView,  CrDeviceSetting_Disable);
                                        :
SCRSDK::SetDeviceSetting(handle, Setting_Key_EnableLiveView, CrDeviceSetting_Enable);
```

In the following sample code, setting 2 will change the captured still image data to be divided into 2MB each and transferred to the host PC.

If you want to return to the initial state, set “CrPartialFile_Default”.

The initial state varies depending on the connection type.

See SetDeviceSetting() for details.

*Example:*

```
SCRSDK::SetDeviceSetting(handle, Setting_Key_PartialBuffer, 2);
                                        :
SCRSDK::SetDeviceSetting(handle, Setting_Key_PartialBuffer, CrPartialFile_Default);
```

# Download and upload setting files

You can save(download) the camera settings as a file on the host PC or a storage device connected to the host PC. You can restore the camera settings by uploading the file saved with this API to the camera. You can only upload to the same model. It is also possible to upload to another camera of the same model.

DownloadSettingFile() has four parameters. The second parameter specifies the type of file to download from the camera. Specify the file save path in the third parameter and the file name in the fourth parameter.

Specify the file save location in the third parameter and the file name in the fourth parameter.

refs. DownloadSettingFile() API

*Example:*

```
CrError err = SCRSDK::DownloadSettingFile(
                    handle,
                    SCRSDK::CrDownloadSettingFileType_Setup,
                    path,
                    name);


                    :
```

UploadSettingFile() has three parameters. The second parameter specifies the type of file to upload to the camera. The third specifies the full path of the file to upload to the camera.

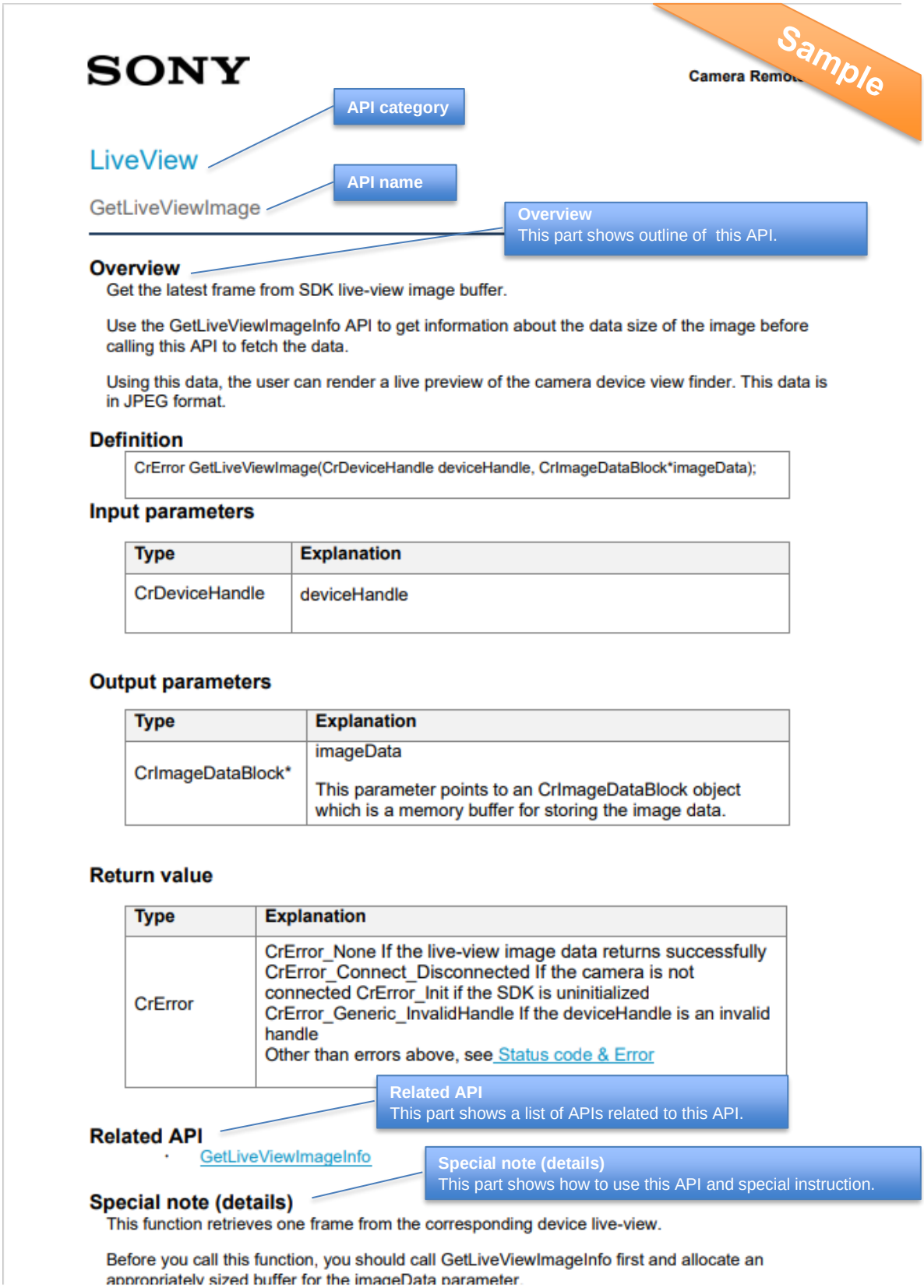
The upload result will be notified by a OnWarning(). If a file for another model or an invalid file is uploaded, CrWarning_CameraSettings_Read_Result_Invalid etc. will be returned.

*Example:*

```
SCRSDK:: UploadSettingFile(
                    handle,
                    SCRSDK::CrUploadSettingFileType_Setup,
                    filepath);


                    :
```

# API Reference

This chapter provides the detailed API specification of Camera Remote SDK using the below format.



# Initialize

Init

## Overview

Initialize the Camera Remote SDK for use. This function must be called before calling any other Camera Remote SDK function.

## Definition

```
bool Init(CrInt32u logtype = 0);
```

## Input Parameters

| Type | Explanation |
|---|---|
| CrInt32u | Logtype. Only 0 is available in this version. |

## Return values

| Type | Explanation |
|---|---|
| bool | Return parameter<br>If initialize successfully, the result is true; otherwise, the result is false. |

## Related API

- Release

## Special note (details)

During Initialization, the Camera remote SDK is modifying the Rounding Control and Precision Control bits of the CPU floating point control word (Rounding is set to Chop, and Precision to 53 bits). If the Windows user application is using floating point calculation, some calculation results may be modified accordingly. Use the round/ceil/floor functions for proper control.

# Release

Release

## Overview

Terminate the Camera Remote SDK by deleting objects and releasing the memory used by the Camera Remote SDK. Use this function to clean up resources when the Camera Remote SDK is no longer required. Should be called after disconnecting all connected cameras and before your application close.

## Definition

```
bool Release();
```

## Input Parameters

Empty.

## Return values

| Type | Explanation |
|---|---|
| bool | Always returns true |

## Related API

- Init

## Special note (details)

None in particular.

# CameraObject

## EnumCameraObjects

### Overview

The API generates a list of “connectable” cameras. Even if a Sony camera is visible to the PC, if the camera doesn’t have PC remote control feature or if the camera doesn’t have compatibility with this version of Camera Remote SDK, the camera is not listed. Please refer the target model list of this Camera Remote SDK.

### Definition

```
CrError EnumCameraObjects(ICrEnumCameraObjectInfo** ppEnumCameraObjectInfo,
CrInt8u timeInSec = 3);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrInt8u | timeInSec<br>This parameter is not supported with the current Camera Remote SDK. |

### Output parameters

| Type | Explanation |
|---|---|
| ICrEnumCameraObjectInfo** | ppEnumCameraObjectInfo<br><br>This is an input/output parameter.<br><br>When this API returns, ppEnumCameraObjectInfo points an enumerator object to enumerate the connected cameras. If this pointer is null, no suitable camera devices were found.<br><br>When the function returns successfully, the new object will be allocated within the function by the SDK. And because this pointer is overwritten in the SDK, calling EnumCameraObjects with unreleased memory object of this parameter will cause of leaking memory. |

### Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Init if the SDK is uninitialized<br>CrError_Adaptor_HandlePlugin if any plugin modules are not found<br>Other than errors above, see Status code & Error |

### Related API

- Connect
- ICrEnumCameraObjectInfo::Release

## Special note (details)

This is a factory function. Release the list by calling ICrEnumCameraObjectInfo::Release interface function.

Enumerates all supported devices which are currently connected to the PC.

If no supported devices are found, ppEnumCameraObjectInfo remains nullptr.

If supported devices are discovered, ppEnumCameraObjectInfo points to the enumerator object. Their related information can be accessed through the ICrEnumCameraObjectInfo interface.

The information obtained through this API is required by the SDK Connect API.

# CreateCameraObjectInfo

## Overview

ICrCameraObjectInfo is an interface to detect a connectable camera that is connected to the PC. It can be retrieved by ICrEnumCameraObjectInfo using GetCameraObjectInfo(), but can be created by calling CreateCameraObjectInfo(). This ICrCameraObjectInfo interface is used when the program connects a camera.

## Definition

```
ICrCameraObjectInfo* CreateCameraObjectInfo(CrChar* name, CrChar* model, CrInt16 usbPid, CrInt32u idType, CrInt32u idSize, CrInt8u* id, CrChar* connectTypeName, CrChar* adaptorName, CrChar* pairingNecessity,  CrInt32u sshSupport = 0);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrChar* | name<br><br>Not available. |
| CrChar* | model<br><br>Null-terminated device model name string |
| CrInt16 | usbPid<br><br>Pid for usb devices |
| CrInt32u | idType<br><br>For PTP_USB, this is CAMERAOBJECTID_TYPE_USB. |
| CrInt32u | idSize<br><br>Size in bytes of the id buffer |
| CrInt8u* | id<br><br>A buffer containing device information |
| CrChar* | connectTypeName<br><br>A char pointer which points to the null-terminated string of the connection type name of the camera.<br><br>For PTP_USB, the string is "USB"; |
| CrChar* | adaptorName<br><br>A char pointer which points to the null-terminated string of the adapter name of the camera.<br><br>For PTP_USB, the string is "Cr_PTP_USB"; |
| CrChar* | pairingNecessity<br><br>Call with NULL, because this parameter is not used. |

| | |
|---|---|
| CrInt32u | sshSupport<br><br>This parameter is optional.<br><br>For SSH authentication models, set CrSSHsupport_ON. |

All input parameter values are obtained from the EnumCameraObjects API. The user must decide how to preserve these values for use by the Connect API.

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| ICrCameraObjectInfo* | A pointer which points to a newly allocated ICrCameraObjectInfo interface object. The allocation is performed internally by the SDK.<br><br>An object of this type is required when calling the Connect API. |

## Related API

- Connect
- EnumCameraObjects
- ICrCameraObjectInfo::Release

## Special note (details)

This is a factory function that returns an ICrCameraObjectInfo* to an object allocated by the SDK. An ICrCameraObjectInfo is required to call the Connect API and connect to the corresponding device.

Remember to release the obtained ICrCameraObjectInfo by calling the ICrCameraObjectInfo::Release() interface function. Do not call delete manually.

# CreateCameraObjectInfoUSBConnection

## Overview

CreateCameraObjectInfoUSBConnection() is an API that creates a “Camera Object” for USB connection camera with the information specified by the user.

The purpose of this API is to create the “Camera Object” required for Connect() without using the EnumCameraObjects() when the target camera has already been determined.

The “Camera Object” obtained as a result of the EnumCameraObjects() and the “Camera Object” obtained by using this API does not exactly match, but there is no problem in operating the target camera.

## Definition

```
CrError CreateCameraObjectInfoUSBConnection(ICrCameraObjectInfo** pCameraObjectInfo,
CrCameraDeviceModelList model, CrInt8u* usbSerialNumber);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrCameraDeviceModelList | model<br><br>Model of the Camera. Use the CrCameraDeviceModelList defined in CrDefines.h. |
| CrInt8u* | usbSerialNumber<br><br>Serial number for usb devices.<br>12byte + Null-terminated<br><br>refs. To check the USB serial number |

## Output parameters

| Type | Explanation |
|---|---|
| ICrCameraObjectInfo** | pCameraObjectInfo<br><br>A pointer to the ICrCameraObjectInfo. Specify the address of a modifiable ICrCameraObjectInfo pointer.<br><br>Caution:<br>pCameraObjectInfo created with information different from the camera you actually want to operate is not guaranteed to be used. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_OutOfModelList CrCameraDeviceModelList If the value does not exist in the model<br>CrError_Api_NotSupportModelOfUSB USB For unsupported model<br>CrError_Api_InvalidSerialNumber If usbSerialNumber is null |

## Related API

- Connect
- EnumCameraObjects
- ICrCameraObjectInfo::Release

## Special note (details)

The pCameraObjectInfo generated by this API does not match the pCameraObjectInfo of the actual camera returned by executing EnumCameraObjects().

It is not considered to use the pCameraObjectInfo returned by EnumCameraObjects() and the pCameraObjectInfo generated by this API at the same time, and the operation in that case is not guaranteed.

## CreateCameraObjectInfoEthernetConnection

### Overview

CreateCameraObjectInfoEthernetConnection() is an API that creates a “Camera Object” for Ethernet connection camera with the information specified by the user.

The purpose of this API is to create the “Camera Object” required for Connect() without using the EnumCameraObjects() when the target camera has already been determined.

The “Camera Object” obtained as a result of the EnumCameraObjects() and the “Camera Object” obtained by using this API does not exactly match, but there is no problem in operating the target camera.

### Definition

```
CrError CreateCameraObjectInfoEthernetConnection(ICrCameraObjectInfo** pCameraObjectInfo, CrCameraDeviceModelList model, CrInt32u ipAddress, CrInt8u* macAddress, CrInt32u sshSupport = 0);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrCameraDeviceModelList | model<br><br>Model of the Camera. Use the CrCameraDeviceModelList defined in CrDefines.h. |
| CrInt32u | ipAddress<br><br>IP address of the camera<br><br>Ex.) 192.168.0.5 = 0x0500A8C0<br><br>To convert a dot-separated string notation to a 32-bit value<br>Please set the 1st <-> 7~0bit, the 2nd <-> 15~8bit, the 3rd <-> 23~16bit, and the 4th <-> 31~24bit. |
| CrInt8u* | macAddress<br><br>MAC address of the camera. 6byte fixed.<br><br>This value is used to identify the “Camera Object”. It is not always necessary to specify the MAC address of the camera body. If you create multiple “Camera Object”, it is recommended to specify different 6-byte data for each. |
| CrInt32u | sshSupport<br><br>This parameter is optional. For SSH authentication models, set CrSSHsupport_ON.<br><br>Caution:<br>Default is CrSSHsupport_OFF. If this parameter is omitted for a camera that requires SSH authentication, connect will fail. |

## Output parameters

| Type | Explanation |
|---|---|
| ICrCameraObjectInfo** | pCameraObjectInfo<br><br>A pointer to the ICrCameraObjectInfo. Specify the address of a modifiable ICrCameraObjectInfo pointer.<br><br>Notice:<br>pCameraObjectInfo created with information different from the camera you actually want to operate is not guaranteed to be used. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_OutOfModelList CrCameraDeviceModelList If the value does not exist in the model<br>CrError_Api_NotSupportModelOfEthernet For unsupported model<br>CrError_Api_InvalidIpAddress If the IP address is judged to be inappropriate<br>CrError_Api_InvalidMacAddress If the MAC address is judged to be inappropriate |

## Related API

- Connect
- EnumCameraObjects
- ICrCameraObjectInfo::Release

## Special note (details)

The pCameraObjectInfo generated by this API does not match the pCameraObjectInfo of the actual camera returned by executing EnumCameraObjects().

It is not considered to use the pCameraObjectInfo returned by EnumCameraObjects() and the pCameraObjectInfo generated by this API at the same time, and the operation in that case is not guaranteed.

## GetFingerprint

### Overview

This API gets a fingerprint data from a camera that requires an SSH authentication connection.

Getting and checking the fingerprint is the only way to avoid connecting to the wrong destination (SSH server other than camera). The user should compare the fingerprint acquired by this API with the fingerprint displayed on the camera body and judge whether it is correct or not. If they do not match, the Connect() will fail even if you proceed to the connection process.

Fingerprint data changes when the camera body is initialized or the fingerprint is regenerated on the camera body. Fingerprint data is required for Connect(), but it does not require GetFingerprint() every time before Connect(). Only when the fingerprint data does not change, the fingerprint data obtained by this API can be used as a parameter of Connect() many times.

### Definition

```
CrError GetFingerprint(ICrCameraObjectInfo* pCameraObjectInfo, char* fingerprint, CrInt32u* fingerprintSize);
```

### Input parameters

| Type | Explanation |
|---|---|
| ICrCameraObjectInfo* | pCameraObjectInfo<br><br>he camera which is going to be connected. This parameter is return by ICrEnumCameraObjectInfo::GetCameraObjectInfo(). |

### Output parameters

| Type | Explanation |
|---|---|
| char* | fingerprint<br><br>The fingerprint pointer. Developer prepares a larger buffer to receive fingerprint data, and passes the address of this pointer.<br>When function returns successfully, this parameter will points to a Base64 encoded character.<br><br>Note:<br>Add the "=" for padding.<br>Does not contain Null-terminations. |
| CrInt32u* | fingerprintSize<br><br>A pointer to an integer which indicates the size of fingerprint data. Developers should pass the address of a modifiable CrInt32u variable. This function will write the size of the returned fingerprint data to the variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidParameter If the parameter is NULL<br>CrError_Connect_SSH_NotSupported If SSH connection is not supported<br>CrError_Connect_SSH_GetFingerprintFailed If for some reason the Fingerprint could not be obtained from the specified camera. The probable reason is that the IP address of the camera object created by CreateCameraObjectInfoEthernetConnection() is incorrect.<br>Other than errors above, see Status code & Error |

## Related API

- Connect
- EnumCameraObjects
- ICrCameraObjectInfo::Release

## Special note (details)

None in particular

# Connection

## Connect

### Overview

This API attempts to connect to the camera device specified by the user.

This function is an asynchronous connection request. If this function returns without error, the asynchronous connection request has been initiated successfully.

Success or failure of the connection is communicated to the user through the IDeviceCallback interface. This interface must be implemented by the user to use the Camera Remote SDK.

The content transfer function has been added from version 1.05.00, and the openMode parameter has been added to this API. The openMode parameter is optional. This can be omitted when performing remote control as before.

From version 1.06.00, the function to specify the behavior of automatic reconnection and the information for SSH authenticate connection has been added. The automatic reconnection control parameters are optional. By default, automatic reconnection is enabled, but when in ContentsTransferMode, automatic reconnection is forcibly disabled. This is due to the limitations of the camera body.

- See “Supporting physical layer” for content transfer support models
- See “Pull out content stored on media” for content transfer capabilities

When operating a camera that requires SSH authentication, it is necessary to set a User name and Password on the camera body. In addition, it is necessary to acquire fingerprint data in advance with GetFingerprint().

### Definition

```
CrError Connect(ICrCameraObjectInfo* pCameraObjectInfo,  IDeviceCallback* callback,
CrDeviceHandle* deviceHandle, CrSdkControlMode openMode = CrSdkControlMode_Remote,
CrReconnectingSet reconnect = CrReconnecting_ON , const char* userId = 0, const char*
userPassword = 0, const char* fingerprint = 0, CrInt32u fingerprintSize = 0);
```

### Input parameters

| Type | Explanation |
|---|---|
| ICrCameraObjectInfo* | pCameraObjectInfo<br><br>The camera which is going to be connected. This parameter is return by ICrEnumCameraObjectInfo::GetCameraObjectInfo(). |
| IDeviceCallback* | callback<br><br>The user-implemented device callback interface. App developers who use this SDK should implement the callback function interface to handle events from the camera such as connected or disconnected, property change, etc. |

| | |
|---|---|
| CrSdkControlMode | openMode<br><br>This parameter is optional. If you want to pull out the contents of the media and save it on the host device, specify "CrSdkControlMode_ContentsTransfer".<br><br>Note :<br>Switching between RemoteControlMode and ContentsTransferMode cannot be performed while connected. After disconnecting in each mode, reconnect in the desired mode. |
| CrReconnectingSet | reconnect<br><br>This parameter is optional. With the default value, the SDK that detects an unexpected disconnection will try to reconnect for a period of time (= called the automatic reconnection function). Specify CrReconnecting_OFF when you want to disable the automatic reconnection function. |
| const char* | userId<br><br>This parameter is optional. Specify the User name for the SSH authentication. Make it null terminated.<br><br>For details on how to set the User name for SSH authentication, refer to the help guide for the target camera. |
| const char* | userPassword<br><br>This parameter is optional. Specify the password for the SSH authentication. Make it null terminated.<br><br>For details on how to set the password for SSH authentication, refer to the help guide for the target camera. |
| const char* | fingerprint<br><br>This parameter is optional. Specify the fingerprint data obtained by GetFingerprint(). |
| CrInt32u | fingerprintSize<br><br>This parameter is optional. Specify the length of the fingerprint parameter. |

## Input/Output parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle* | deviceHandle<br><br>The handle of the connected camera is returned in the variable. This must be set 0 before calling Connect(). |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_Unknown If the pCameraObjectInfo is NULL, and no valid deviceNumber is supplied<br>CrError_Connect_ContentsTransfer_NotSupported Connected to a model that does not support content transfer.<br>Errors starting with CrError_Connect_SSH, such as CrError_Connect_SSH_ServerConnectFailed, indicate an SSH connection error.<br>Other than errors above, see Status code & Error |

## Related API

- GetFingerprint
- Disconnect
- EnumCameraObjects
- CreateCameraObjectInfo
- IDeviceCallback::OnConnected

## Special note (details)

This API can be used in two ways: to connect to a new device and to reconnect to an existing device.

To connect to a new device, supply a deviceHandle value of 0 and a pointer to a valid ICrCameraObjectInfo.

To reconnect to an existing device, supply the deviceHandle of that device to this API and NULL in pCameraObjectInfo. The SDK will then reuse the existing internal device handle and attempt to connect to the specified camera device. Reconnection will not work if the specific device was previously released with the ReleaseDevice API. In this case, CrError_Generic_Unknown will be returned.

A successful connection is reported to the user through the IDeviceCallback::OnConnected interface function. An implementation of this function must be supplied to Connect by the user though the callback parameter.

The deviceHandle out-parameter returns the SDK device identifier to the user. This identifier is required to use subsequent SDK API functions to interact with the connected device.

Repeatedly entering the wrong SSH parameters will lock the camera. In that case, turn off the power switch of the camera and restart it, or wait for a while and then try again.

# Disconnect

## Overview

This API function disconnects the indicated device.

After calling this API, the deviceHandle remains valid and can be used to reconnect to the same device.

## Definition

```
CrError Disconnect(CrDeviceHandle deviceHandle);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the deviceHandle is a valid handle. In this case, the connection to the camera will be closed.<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- Connect
- ReleaseDevice
- IDeviceCallback::OnDisconnected

## Special note (details)

Stops the internal processing threads on the indicated device and disconnects from the device.

Calling this API will not invalidate the existing deviceHandle. This function simple disconnects the device. Unless ReleaseDevice is called, the device handle can be reused to connect to the same device.

The SDK signals successful disconnection by calling IDeviceCallback::OnDisconnected.

# Device

## ReleaseDevice

### Overview

This API requests that the SDK release the resources allocated for the specified device.

Calling this API will invalidate the provided deviceHandle. Do not attempt to reuse it after calling this API.

### Definition

```
CrError ReleaseDevice(CrDeviceHandle deviceHandle);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

### Output parameters

None

### Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the deviceHandle is a valid handle.<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

### Related API

- Connect
- Disconnect
- IDeviceCallback::OnDisconnected

### Special note (details)

This function releases the resources associated with the specified device handle.

# Device Property

## GetDeviceProperties

### Overview

This API gets device properties from the device specified by the deviceHandle.

This retrieves all of the available properties of device. This list contains information about each property's current value, list of valid values and whether or not the property value can currently be updated by the user.

### Definition

```
CrError GetDeviceProperties(CrDeviceHandle deviceHandle, CrDeviceProperty** properties, CrInt32* numOfProperties);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

### Output parameters

| Type | Explanation |
|---|---|
| CrDeviceProperty** | properties<br><br>The property list pointer. Developers should pass the address of a modifiable CrDeviceProperty pointer. The value of this pointer should be initialized to nullptr.<br><br>The function will make a copy of the SDK-internal CrDeviceProperty list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of CrDeviceProperty list. |
| CrInt32* | numOfProperties<br><br>A pointer to an integer which indicates the number of CrDeviceProperty objects in the property list.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to the variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the properties are returned successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetSelectDeviceProperties
- ReleaseDeviceProperties
- SetDeviceProperty
- IDeviceCallback::OnPropertyChanged
- IDeviceCallback::OnPropertyChangedCodes

## Special note (details)

This is a factory function. The SDK will allocate memory. Call the ReleaseDeviceProperties API to correctly release the generated list.

This API function retrieves a list of all the properties supported by the indicated device. Each returned property also provides its current value, a list of values it supports and whether or not the property is currently modifiable.

It is important to initialize the out-parameter pointer to nullptr before passing it to this function. This is required to detect whether or not a list has been created. The out-parameter properties will remain unmodified if the property list cannot be retrieved.

If the list is successfully retrieved, properties points to the list and out-parameter numOfProperties indicates the number of items in the list.

## GetSelectDeviceProperties

### Overview

This API gets specified device properties from the device specified by the deviceHandle.

This list contains information about each property's current value, list of valid values and whether or not the property value can currently be updated by the user.

### Definition

```
CrError GetSelectDeviceProperties(CrDeviceHandle deviceHandle, CrInt32u numOfCodes,
CrInt32u* codes, CrDeviceProperty** properties, CrInt32* numOfProperties);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrInt32u | numOfCodes<br><br>Number of device properties to get. |
| CrInt32u* | codes<br><br>List of device property codes to get. |

### Output parameters

| Type | Explanation |
|---|---|
| CrDeviceProperty** | properties<br><br>The property list pointer. Developers should pass the address of a modifiable CrDeviceProperty pointer. The value of this pointer should be initialized to nullptr.<br><br>The function will make a copy of the SDK-internal CrDeviceProperty list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of CrDeviceProperty list. |
| CrInt32* | numOfProperties<br><br>A pointer to an integer which indicates the number of CrDeviceProperty objects in the property list.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to the variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the properties are returned successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetDeviceProperties
- ReleaseDeviceProperties
- SetDeviceProperty
- IDeviceCallback::OnPropertyChangedCodes

## Special note (details)

This is a factory function. The SDK will allocate memory. Call the ReleaseDeviceProperties API to correctly release the generated list.

This API function retrieves a list of specified properties supported by the indicated device. Each returned property also provides its current value, a list of values it supports and whether or not the property is currently modifiable.

It is important to initialize the out-parameter pointer to nullptr before passing it to this function. This is required to detect whether or not a list has been created. The out-parameter properties will remain unmodified if the property list cannot be retrieved.

If the list is successfully retrieved, properties points to the list and out-parameter numOfProperties indicates the number of items in the list.

# ReleaseDeviceProperties

## Overview
This API function releases the CrDeviceProperty list allocated by GetDeviceProperties.

## Definition
```
CrError ReleaseDeviceProperties(CrDeviceHandle deviceHandle, CrDeviceProperty* properties);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDeviceProperty* | properties<br><br>The property list pointer pointing to the list to be released. |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the property list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetDeviceProperties
- GetSelectDeviceProperties

## Special note (details)
This function releases the CrDeviceProperty list that is associated with the specified device handle.

# SetDeviceProperty

## Overview

Request the SDK set a new value to the selected property for the corresponding device.

The function is asynchronous and returns to the user as soon as the SDK enqueues the requested action. After the property of the camera changed, OnPropertyChanged() and other callback functions are called and GetDeviceProperties() will return the new property value.

## Definition

```
CrError SetDeviceProperty(CrDeviceHandle deviceHandle, CrDeviceProperty* pProperty);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDeviceProperty* | pProperty<br><br>This parameter points to the CrDeviceProperty object which contains the property that will be set to the device. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the command is sent out.<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetDeviceProperties
- GetSelectDeviceProperties
- IDeviceCallback::OnPropertyChanged
- IDeviceCallback::OnPropertyChangedCodes

## Special note (details)

Requests the SDK set the indicated pProperty on the corresponding device indicated by deviceHandle.

pProperty contains the desired property code and desired property value.

The desired value should be one of the valid values retrieved from GetDeviceProperties. The SDK will not set an unsupported value.

The return value from this function will not indicate whether or not the property was set successfully. If the property is updated successfully the SDK will call IDeviceCallback:: OnPropertyChanged() and other callback functions. The warning code will indicate that a property has changed.

# Send Command

SendCommand

## Overview

This API function sends commands for controlling the device. This allows the user to control camera functions such as the shutter release. When stopping continuous shooting, use “CrCommandId_Release” with “CrCommandParam_Up”.

The function is asynchronous and returns to the user as soon as the SDK enqueues the requested action. The effects of sending a command can be confirmed by observing the actual device for the requested change.

## Definition

```
CrError SendCommand(CrDeviceHandle deviceHandle, CrInt32u commandId,
CrCommandParam commandParam);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrInt32u | commandId<br><br>This parameter is one of CrCommandId defined in CrCommandData.h. |
| CrCommandParam | commandParam<br><br>This parameter is one of CrCommandParam defined in CrCommandData.h. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the command is sent out.<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- SetDeviceProperty

## Special note (details)

Requests the SDK send a command to the device indicated by deviceHandle.
The command to send is identified by commandId.

# LiveView

## GetLiveViewImage

### Overview

Get the latest frame from SDK live-view image buffer.

Use the GetLiveViewImageInfo API to get information about the data size of the image before calling this API to fetch the data.

Using this data, the user can render a live preview of the camera device view finder. This data is in JPEG format.

### Definition

```
CrError GetLiveViewImage(CrDeviceHandle deviceHandle, CrImageDataBlock*imageData);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

### Output parameters

| Type | Explanation |
|---|---|
| CrImageDataBlock* | imageData<br><br>This parameter points to an CrImageDataBlock object which is a memory buffer for storing the image data. |

### Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the live-view image data returns successfully<br>CrError_Connect_Disconnected If the camera is not connected CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

### Related API

- GetLiveViewImageInfo

### Special note (details)

This function retrieves one frame from the corresponding device live-view.

Before you call this function, you should call GetLiveViewImageInfo first and allocate an appropriately sized buffer for the imageData parameter.

This function does not send or receive any data from the device but merely copy the live image data from a buffer, the buffer is updated in real time by background task.

# GetLiveViewImageInfo

## Overview
This function returns the data size of the live-view image.

## Definition
```
CrError GetLiveViewImageInfo(CrDeviceHandle deviceHandle, CrImageInfo* info);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

## Output parameters

| Type | Explanation |
|---|---|
| CrImageInfo* | info<br><br>This parameter points to a CrImageInfo object. If function returns successfully, the member bufferSize of the CrImageInfo object will be set appropriately according to the live-view image settings. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the CrImageInfo is properly set CrError_Connect_Disconnected If the camera is not connected CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetLiveViewImage

## Special note (details)
This function is used to retrieve the size of the live-view image. Use the retrieved value to create a buffer to store the live-view image.

Call this function prior to calling GetLiveViewImage.

## GetLiveViewProperties

### Overview

Get live view properties from the specified device. Functionally equivalent to GetProperties for properties related to the device live-view.

The properties retrieved by this API call are closely related to the camera live-view image. These properties are not included in the list of properties retrieved by GetDeviceProperties.

### Definition

```
CrError GetLiveViewProperties(CrDeviceHandle deviceHandle, CrLiveViewProperty** properties, CrInt32* numOfProperties);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

### Output parameters

| Type | Explanation |
|---|---|
| CrLiveViewProperty** | properties<br><br>The property list pointer. Developers should pass the address of a modifiable CrLiveViewProperty pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the SDK-internal CrLiveViewProperty list for the indicated deviceHandle.<br>When function returns successfully, this parameter will point to the copy of CrLiveViewProperty list.<br>Must be freed with ReleaseLiveViewProperties() after use. |
| CrInt32* | numOfProperties<br><br>A pointer to an integer which indicates the number of CrLiveViewProperty objects in the property list.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to this location. |

### Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetSelectLiveViewProperties
- ReleaseLiveViewProperties
- IDeviceCallback::OnLvPropertyChanged
- IDeviceCallback::OnLvPropertyChangedCodes

## Special note (details)

This is a factory function. The SDK will allocate memory if required.

This API function retrieves a list of all the live-view properties supported by the indicated device. Each returned property also provides its current value, a list of values it supports and whether or not the property is currently modifiable.

The out-parameter properties will remain unmodified if the property list cannot be retrieved.

If the list is successfully retrieved, properties points to the list and out-parameter numOfProperties indicates the number of items in the list.

## GetSelectLiveViewProperties

### Overview

Get specified live view properties from the specified device. Functionally equivalent to GetSelectDeviceProperties for properties related to the device live-view.

The properties retrieved by this API call are closely related to the camera live-view image. These properties are not included in the list of properties retrieved by GetDeviceProperties or GetSelectDeviceProperties.

### Definition

```
CrError GetSelectLiveViewProperties(CrDeviceHandle deviceHandle, CrInt32u numOfCodes, CrInt32u* codes, CrLiveViewProperty** properties, CrInt32* numOfProperties);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrInt32u | numOfCodes<br><br>Number of live-view properties to get. |
| CrInt32u* | codes<br><br>List of live-view property codes to get. |

### Output parameters

| Type | Explanation |
|---|---|
| CrLiveViewProperty** | properties<br><br>The property list pointer. Developers should pass the address of a modifiable CrLiveViewProperty pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the specified CrLiveViewProperty list for the indicated deviceHandle.<br>When function returns successfully, this parameter will point to the copy of CrLiveViewProperty list.<br>Must be freed with ReleaseLiveViewProperties() after use. |
| CrInt32* | numOfProperties<br><br>A pointer to an integer which indicates the number of CrLiveViewProperty objects in the property list.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to this location. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetLiveViewProperties
- ReleaseLiveViewProperties
- IDeviceCallback::OnLvPropertyChangedCodes

## Special note (details)

This is a factory function. The SDK will allocate memory if required.

This API function retrieves a list of all the live-view properties supported by the indicated device. Each returned property also provides its current value, a list of values it supports and whether or not the property is currently modifiable.

The out-parameter properties will remain unmodified if the property list cannot be retrieved.

If the list is successfully retrieved, properties points to the list and out-parameter numOfProperties indicates the number of items in the list.

# ReleaseLiveViewProperties

## Overview
This API function releases the CrLiveViewProperty list allocated by GetLiveViewProperties.

## Definition
```
CrError ReleaseLiveViewProperties(CrDeviceHandle deviceHandle, CrLiveViewProperty* properties);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrLiveViewProperty* | properties<br><br>The live-view property list pointer pointing to the list to be released. |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetLiveViewProperties
- GetSelectLiveViewProperties

## Special note (details)
Allows the SDK to release the SDK-allocated memory for the corresponding device live-view properties list.

Supply a connected device handle.

# Device Setting

## GetDeviceSetting

### Overview

This function returns SDK settings for the specified device.

Please check SetDeviceSetting() for gettable information.

### Definition

```
CrError GetDeviceSetting(CrDeviceHandle deviceHandle, CrInt32u key, CrInt32u* value);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrInt32u | key<br><br>Key for the setting to retrieve. Values can be found in the SettingKey enumeration. |

### Output parameters

| Type | Explanation |
|---|---|
| CrInt32* | value<br><br>The current value of the key in question.<br><br>App developers should pass the address of a modifiable CrInt32 object. This function will write the current value of the key of interest here. |

### Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error. |

### Related API

- SetDeviceSetting

### Special note (details)

None in particular

# SetDeviceSetting

## Overview
This API updates SDK settings for the indicated device.

## Definition

```
CrError SetDeviceSetting(CrDeviceHandle deviceHandle, CrInt32u key, CrInt32u value);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrInt32u | key<br><br>Key for the setting to update. The following keys can be set.<br>- Setting_Key_EnableLiveView<br>Controls whether or not the Live View image capturing function is enabled.<br>- Setting_Key_PartialBuffer<br>Controls the data transfer size when saving captured images to the host PC. |
| CrInt32u | value<br><br>The new value for key. Refer to the Special note for the values that can be set for each key. |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetDeviceSetting

## Special note (details)

- Setting_Key_EnableLiveView

  Controls whether or not the Live View image capturing function is enabled.
  Use the CrDeviceSetting defined in CrDefines.h.
  The default value is "Enable". Note that if GetLiveViewImage() is called after the change to "Disable", the Live View image retrieved from the camera immediately before the change to "Disable" will be returned.

- Setting_Key_PartialBuffer

  Controls the data transfer size when saving captured  images to the host PC.
  Use the CrPartialFile defined in CrDefines.h.
  Normally (i.e., by CrPartialFile_Default), when connected via Ethernet, one captured image is transferred in 1 MB increments and stored on the host PC; when connected via USB, one captured image is transferred as a batch and stored on the host PC.
  The advantage of partial transfer via USB connection is that SDK processing is not dominated by the transfer of captured images.
  Live View images are designed to be displayed at a maximum of 30 fps, but if the batch transfer of captured image data takes time, the interval between Live View image acquisition processes will increase, resulting in a drop in fps. If the Live View image does not display smoothly when shooting via USB connection, consider adjusting the transfer size in this setting. However, please note that the smaller the specified size, the longer the total time required to complete the transfer of one captured image.

# SetSaveInfo

## Overview

This function sets the location on the PC for saving images transferred from the device.

See Change the Store Image Folder and the File Name for how to use this API function

## Definition

```
CrError SetSaveInfo(CrDeviceHandle deviceHandle, CrChar* path, CrChar* prefix, CrInt32 no);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrChar* | path<br><br>The local path where images should be saved.<br><br>This path is also a content transfer path.<br>If you do not specify the filePath parameter of the PullContentsFile function, the path specified by this parameter is used.<br>If an invalid path is specified for this parameter, normal operation of image transfer in Remote Control Mode and content transfer in Contents Transfer Mode cannot be guaranteed. |
| CrChar* | prefix<br><br>The prefix to give saved images.<br><br>This parameter is valid only when shooting in RemoteControlMode. Not used in ContentsTransferMode. |
| CrInt32 | no<br><br>The starting value to use when enumerating images.<br><br>This parameter is valid only when shooting in RemoteControlMode. Not used in ContentsTransferMode. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the function returns successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- PullContentsFile

## Special note (details)

The save path should be set to a location for which the application has writing access.

# SDK Version

GetSDKVersion

## Overview

This function returns the SDK version number.

## Definition

```
CrInt32u GetSDKVersion();
```

## Input parameters

None

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrInt32u | The SDK Version is represented as a 4-byte unsigned integer constant.<br><br>The first 3 bytes contain the SDK version. The last byte is reserved by the SDK for future use. |

### Error Codes

No Error

## Related API

- GetSDKSerial

## Special note (details)

The SDK version number is set at build time.
This version number will be updated if the SDK API is changed.

# SDK Serial Number

GetSDKSerial

## Overview

This function returns the SDK serial number.

## Definition

```
CrInt32u GetSDKSerial();
```

## Input parameters

None

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrInt32u | The SDK Serial is represented as a 4-byte unsigned integer constant.<br><br>The last 2 bytes contain the SDK serial. The first 2 byte is reserved by the SDK for future use. |

### Error Codes

No Error

## Related API

- GetSDKVersion

## Special note (details)

The SDK serial number is set at build time.

# Update SDK Information

EditSDKInfo

## Overview
Edit the information about the SDK stored in the config file.

## Definition

```
CrError EditSDKInfo(CrInt16u infotype);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrInt16u | A constant that means the information to update.<br>The constant values are in the SDKInfoType enumeration.<br><br>It is possible to delete camera-specific information with the following values.<br>*SDKINFO_AUTHINFO* |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_Insufficient if the update fails |

## Related API
None

## Special note (details)
None in particular

# Contents Transfer

## GetDateFolderList

### Overview

Gets date folder list from the device specified by the deviceHandle.
This function is the first function to call when pulling out the content in the camera.

### Definition

```
CrError GetDateFolderList(CrDeviceHandle deviceHandle, CrMtpFolderInfo** folders,
CrInt32u* numOfFolders);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

### Output parameters

| Type | Explanation |
|---|---|
| CrMtpFolderInfo** | folders<br><br>The date folder list pointer. Developers should pass the address of a modifiable CrMtpFolderInfo pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the SDK-internal date folder list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of date folder list.<br><br>The date folder list in the SDK is created by retrieving data from the camera only when the developer calls this function. Therefore, it may take some time to return to the first function call. This can be especially time consuming if you have a large number of date folders. |
| CrInt32u* | numOfFolders<br><br>A pointer to an integer which indicates the number of CrMtpFolderInfo objects in the date folder list.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to the variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>However, in the case of blank media, CrError_None is returned but numOfFolders becomes zero, so it is necessary to check numOfFolders at the same time.<br>CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function.<br>Other than errors above, see Status code & Error |

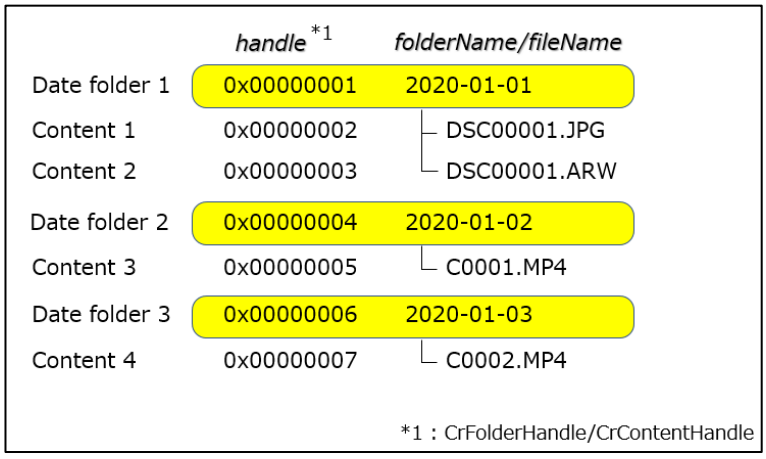
## Related API

- ReleaseDateFolderList
- GetContentsHandleList

## Special note (details)

The date folder information that can be obtained with this API is the handle and folderName in the yellow frame in the figure below.

Fig. If the media has 3 date folders and 4 contents




See Pull out content stored on media for how to use this API function

GetContentsHandleList

## Overview

Gets a handle list of the contents in the date folder specified by folderHandle.

## Definition

```
CrError GetContentsHandleList(CrDeviceHandle deviceHandle, CrFolderHandle folderHandle,
CrContentHandle** contentsHandles, CrInt32u* numOfContents);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrFolderHandle | folderHandle<br><br>Specifies one of the date folder handles obtained by the GetDateFolderList function. |

## Output parameters

| Type | Explanation |
|---|---|
| CrContentHandle ** | contentsHandles<br><br>The content handle list pointer. Developers should pass the address of a modifiable CrContentHandle pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the SDK-internal content handle list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of content handle list.<br><br>The content handle list in the SDK is created by retrieving data from the camera only when the developer calls this function. Therefore, it may take some time to return to the first function call. This can be especially time consuming if you have a large number of content. |
| CrInt32u* | numOfContents<br><br>A pointer to an integer which indicates the number of content in the date folder.<br><br>App developers should pass the address of a modifiable CrInt32 variable. This function will write the size of the returned list to the variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function.<br>Other than errors above, see Status code & Error |

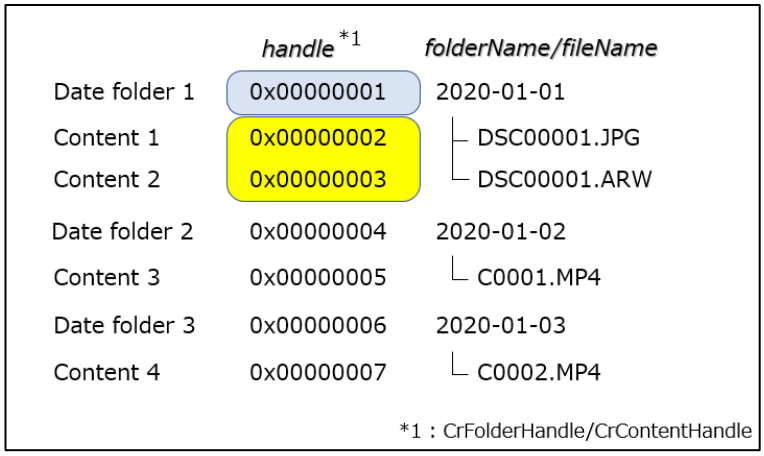
## Related API

- ReleaseContentsHandleList
- GetDateFolderList
- GetContentsDetailInfo

## Special note (details)

For the folderHandle of this API, use one of the date folder handles obtained by GetDateFolderList. Specify handle of blue frame for the folderHandle parameter of this API, you can get the two handles in the yellow frame.

Fig. If the media has 3 date folders and 4 contents




See Pull out content stored on media for how to use this API function

GetContentsDetailInfo

## Overview

Gets a content detail information of the contents specified by contentHandle.

## Definition

```
CrError GetContentsDetailInfo(CrDeviceHandle deviceHandle, CrContentHandle contentHandle, CrMtpContentsInfo* contentsInfo);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrContentHandle | contentHandle<br><br>Specifies one of the content handles obtained by the GetContentsHandleList function. |

## Output parameters

| Type | Explanation |
|---|---|
| CrMtpContentsInfo* | contentsInfo<br><br>The content detail information pointer. Developers should pass the address of a modifiable CrMtpContentsInfo pointer. The value of this pointer should be initialized to nullptr.<br><br>The function will make a copy of the SDK-internal content detail information for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of content detail information.<br><br>The content detail information in the SDK is created by retrieving data from the camera only when the developer calls this function. Therefore, it may take some time to return to the first function call. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Contents_InvalidHandle If the content handle specified is invalid<br>CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function.<br>Other than errors above, see Status code & Error |

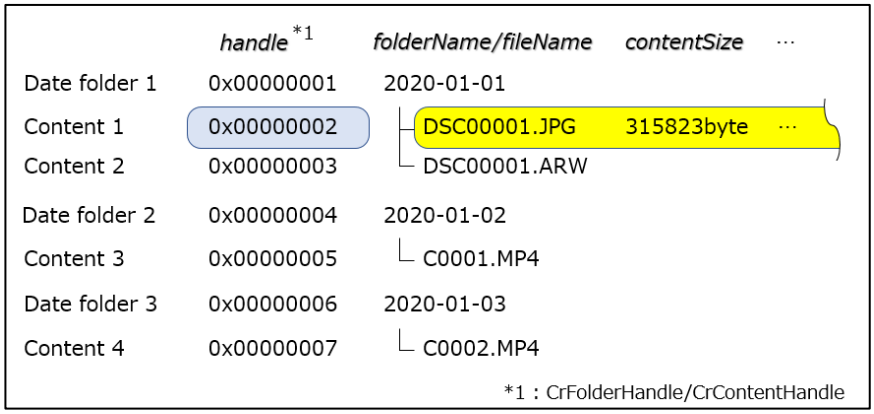
## Related API

- GetContentsHandleList
- PullContentsFile
- GetContentsThumbnailImage

## Special note (details)

For the contentHandle of this API, use one of the content handles obtained by GetContentsHandleList.
You can get the details of the yellow frame by specifying the handle of the blue frame for the contentHandle parameter of this API.

Fig. If the media has 3 date folders and 4 contents




See Pull out content stored on media for how to use this API function

# ReleaseDateFolderList

## Overview

This function releases the CrMtpFolderInfo allocated by GetDateFolderList.

It is not necessary to call this API when zero is returned in the number of folders in GetDateFolderList. Use this API when the number of folders is one or more.

## Definition

```
CrError ReleaseDateFolderList(CrDeviceHandle deviceHandle, CrMtpFolderInfo* folders);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrMtpFolderInfo* | folders<br><br>Date folder list pointer to release. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the date folder list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetDateFolderList

## Special note (details)

None in particular

# ReleaseContentsHandleList

## Overview
This function releases the CrContentHandle array allocated by GetContentsHandleList.

## Definition
```
CrError ReleaseContentsHandleList(CrDeviceHandle deviceHandle, CrContentHandle* contentsHandles);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrContentHandle* | contentsHandles<br><br>Content handle list pointer to release. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the content handle list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetContentsHandleList

## Special note (details)
None in particular

# PullContentsFile

## Overview

Pull contents from the camera. Save a copy of the content file on your host PC.

## Definition

```
CrError PullContentsFile(CrDeviceHandle deviceHandle, CrContentHandle contentHandle,
CrPropertyStillImageTransSize size = CrPropertyStillImageTransSize_Original, CrChar* path =
0, CrChar* fileName = 0);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrContentHandle | contentHandle<br>Specifies one of the content handles obtained by the GetContentsHandleList function.<br>Only content whose details have been obtained using the GetContentsDetailInfo function can be specified. |
| CrPropertyStillImageTransSize | size<br>Specify the size of the still image to be acquired.<br>Specify either CrPropertyStillImageTransSize_Original or CrPropertyStillImageTransSize_SmallSize.<br>When CrPropertyStillImageTransSize_SmallSize is specified<br>You can get a small size image according to the type of still image. JPEG format for JPEG content and HEIF format for HEIF content.<br>If CrDeviceProperty_FileType at the time of shooting is CrFileType_RawJpeg, it will be in JPEG format, and if it is CrFileType_RawHeif, it will be in HEIF format.<br>If you specify small for the movie, an error is returned. |
| CrChar* | path<br>This parameter is optional.<br>If not specified, the path specified in the second parameter of SetSaveInfo will be used. To do this, use SetSaveInfo to change the save destination path in advance. If a path that does not exist in this parameter is specified, or if this parameter is not specified and SetSaveInfo is not used, normal operation of content transfer cannot be guaranteed. |

| | |
|---|---|
| CrChar* | fileName<br><br>This parameter is optional.<br>If not specified, the content will be saved with the file name.<br>If the file name conflicts with an existing file, an additional number is appended after the file name like DSC01234(1).JPG. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Contents_RejectRequest If content cannot be transferred<br>CrError_Generic_NotSupported<br>CrPropertyStillImageTransSize_SmallSize specified for movie content<br>CrError_File_StorageFull Insufficient storage capacity on the host<br>Other than errors above, see Status code & Error |

## Related API

- GetContentsDetailInfo
- GetContentsThumbnailImage
- IDeviceCallback::OnNotifyContentsTransfer
- SetSaveInfo

## Special note (details)

This API cannot guarantee the transfer of content taken with other cameras.
Large files may not be handled depending on the OS.

# GetContentsThumbnailImage

## Overview

Get thumbnail image data.

## Definition

```
CrError GetContentsThumbnailImage(CrDeviceHandle deviceHandle, CrContentHandle contentHandle, CrImageDataBlock* imageData, CrFileType* fileType);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrContentHandle | contentHandle<br>Specifies one of the content handles obtained by the GetContentsHandleList function.<br>Only content whose details have been obtained using the GetContentsDetailInfo function can be specified. |

## Output parameters

| Type | Explanation |
|---|---|
| CrImageDataBlock* | imageData<br>This parameter points to an CrImageDataBlock object which is a memory buffer for storing the image data.<br>JPEG image data of 160 x 120 pixels is set in the pointer.<br>The usage of the CrImageDataBlock class is the same as the GetLiveViewImage function. See LiveView for the size of the buffer to prepare in advance. |
| CrFileType* | fileType<br>A type that means the format of a thumbnail image.<br>Developers should pass the address of a modifiable CrFileType variable.<br>Thumbnail images of JPEG content, RAW content, and movie content are in JPEG format.<br>Thumbnail images of HEIF content are in HEIF format.<br>Caution:<br>For ILCE-1 and ILCE-7SM3 only, the thumbnail image of the RAW content when CrDeviceProperty_FileType is set to CrFileType_RawHeif will be in HEIF format. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Contents_RejectRequest When content is being transferred or thumbnail image data cannot be obtained<br>Other than errors above, see Status code & Error |

## Related API

- GetContentsHandleList
- GetContentsDetailInfo
- PullContentsFile

## Special note (details)

None in particular

# Display string

RequestDisplayStringList

## Overview

You can use this API and GetDisplayStringTypes() and GetDisplayStringList() to get the menu string and menu information displayed on the camera body.

- See to “Get the menu display string” for details.

## Definition

```
CrError RequestDisplayStringList(CrDeviceHandle deviceHandle, CrDisplayStringType type);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDisplayStringType | type<br><br>Specify the type of DisplayStringList you want to get. Use the CrDisplayStringType defined in CrDeviceProperty.h.<br><br>The type of DisplayStringList that can be obtained depends on each model. Not all types are available. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_Insufficient if the update fails |

## Related API

- GetDisplayStringTypes
- GetDisplayStringList
- IDeviceCallback::OnWarning

## Special note (details)

The result will be notified by OnWarning().
If CrWarning_RequestDisplayStringList_Success is notified by OnWarning(), GetDisplayStringTypes() and GetDisplayStringList() will be available.
If CrWarning_RequestDisplayStringList_Error is notified by OnWarning(), the camera may not support the specified type.

# GetDisplayStringTypes

## Overview

This API is used to know the type and number of information acquired by RequestDisplayStringList().

- See to “Get the menu display string” for details.

## Definition

```
CrError GetDisplayStringTypes(CrDeviceHandle deviceHandle, CrDisplayStringType** types,
CrInt32u* numOfTypes);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

## Output parameters

| Type | Explanation |
|---|---|
| CrDisplayStringType** | types<br><br>The CrDisplayStringType list pointer. Developers should pass the address of a modifiable CrDisplayStringType pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the SDK-internal CrDisplayStringType list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of CrDisplayStringType list.<br><br>Note:<br>It may contain CrDisplayStringType that the SDK does not support. |
| CrInt32u* | numOfTypes<br><br>An integer pointer that indicates the number of CrDisplayStringType returned by the types pointer.<br><br>Developers should pass the address of a modifiable CrInt32u variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>However, if numOfTypes is zero, even if CrError_None is returned, it should be judged as fail.<br><br>CrError_Api_NoApplicableInformation The reason why numOfTypes is returned as zero is probably because RequestDisplayStringList() has not been executed yet, or the camera itself does not own the CrDisplayStringType specified by the type parameter of RequestDisplayStringList(). |

## Related API

- RequestDisplayStringList
- GetDisplayStringList
- ReleaseDisplayStringTypes

## Special note (details)

This API is not mandatory. If the processing result of RequestDisplayStringList() is successful, you can call GetDisplayStringList() directly.

# GetDisplayStringList

## Overview

This API gets the menu string and menu information displayed on the camera body.

- See to “Get the menu display string” for details.

## Definition

```
CrError GetDisplayStringList(CrDeviceHandle deviceHandle, CrDisplayStringType type,
CrDisplayStringListInfo** list, CrInt32u* numOfList);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDisplayStringType | type<br><br>Specify the type of DisplayStringList you want to get. Use the CrDisplayStringType defined in CrDeviceProperty.h. The type of DisplayStringList that can be obtained depends on each model. Not all types are available.<br><br>It is recommended to get a list of types that can be referred by GetDisplayStringTypes() in advance and check if the type you want to use exists in it. |

## Output parameters

| Type | Explanation |
|---|---|
| CrDisplayStringListInfo ** | list<br><br>The CrDisplayStringListInfo list pointer. Developers should pass the address of a modifiable CrDisplayStringListInfo pointer. The value of this pointer should be initialized to nullptr.<br>The function will make a copy of the SDK-internal CrDisplayStringListInfo list for the indicated deviceHandle. When the function returns successfully, this parameter will point to the copy of CrDisplayStringListInfo list. Only the information that matches the type specified in the type parameter is copied.<br><br>Note:<br>If CrDisplayStringType_AllList is specified as an input parameter, CrDisplayStringListInfo of CrDisplayStringType that SDK does not support may be returned in the output parameter. |

| | |
|---|---|
| CrInt32u* | numOfList<br><br>An integer pointer that indicates the number of CrDisplayStringListInfo returned by the list pointer.<br><br>Developers should pass the address of a modifiable CrInt32u variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_Insufficient if the update fails |

## Related API

- RequestDisplayStringList
- GetDisplayStringTypes
- ReleaseDisplayStringList

## Special note (details)

When the menu character string or menu information is updated, it will be notified by OnWarning().
refs. CrWarning_DisplayListChanged_Button_AssignDisplayList and more

If the beginning of the warning code of the received warning is "CrWarning_DisplayListChanged_", it is also possible to directly acquire the menu information with this API without checking using RequestDisplayStringList().

# ReleaseDisplayStringTypes

## Overview
This function releases the CrDisplayStringType allocated by GetDisplayStringTypes().

## Definition
```
CrError ReleaseDisplayStringTypes(CrDeviceHandle deviceHandle, CrDisplayStringType* types);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDisplayStringType * | types<br>CrDisplayStringType list pointer to release. |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the Type list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetDisplayStringTypes

## Special note (details)
None in particular

# ReleaseDisplayStringList

## Overview
This function releases the CrDisplayStringListInfo allocated by GetDisplayStringList().

## Definition

```
CrError ReleaseDisplayStringList(CrDeviceHandle deviceHandle, CrDisplayStringListInfo* list);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDisplayStringListInfo * | list<br><br>CrDisplayStringListInfo list pointer to release. |

## Output parameters
None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetDisplayStringList

## Special note (details)
None in particular

# Setting file

## DownloadSettingFile

### Overview

Save (download) the camera settings as a file on the host PC or a storage device connected to the host PC.

By uploading the file saved by this API to the camera with UploadSettingFile(), it is also possible to restore the camera settings.

Before executing this API, please make sure that the media is inserted in the slot of the camera. This is due to the specifications of the camera.

### Definition

```
CrError DownloadSettingFile(CrDeviceHandle deviceHandle, CrDownloadSettingFileType type,
CrChar* filePath = 0, CrChar* fileName = 0);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrDownloadSettingFileType | type<br><br>Specifies the type of file to download. |
| CrChar* | filePath<br><br>This parameter is optional.<br>If not specified, the path specified in the second parameter of SetSaveInfo will be used. To do this, use SetSaveInfo to change the save destination path in advance. If a path that does not exist in this parameter is specified, or if this parameter is not specified and SetSaveInfo is not used, there is no guarantee that this API will be successful. |
| CrChar* | fileName<br><br>This parameter is optional. The extension is fixed to "DAT".<br>If this parameter is not specified, the file will be saved with "Camera model name + _CUMSET.DAT" fixed.<br>If the file name conflicts with an existing file, an additional number is appended after the file name like ILCE-1_CUMSET(1).DAT. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_File_StorageFull  Insufficient storage capacity on the host.<br>Other than errors above, see Status code & Error |

## Related API

- IDeviceCallback::OnCompleteDownload
- IDeviceCallback::OnWarning
- UploadSettingFile

## Special note (details)

This API can be executed when CrDeviceProperty_CameraSetting_SaveOperationEnableStatus is Enable.

The result will be notified by OnWarning() or OnCompleteDownload().
If the save is successful, the file name and file type saved by OnCompleteDownload() will be notified.
If saving fails, OnWarning() will notify you of the cause of the failure.

This API does not support all models.

Saving the setting file can also be realized by operating the camera body without using the API. In that case, the save destination of the file is the "memory card" inserted in the media slot of the camera body.

For ILCE-1 : MENU > Setup > Reset/Set Settings > Save/Load Settings > Save

# UploadSettingFile

## Overview

It is possible to upload the setting file saved in the host PC etc. with DownloadSettingFile() to the camera with this API and restore the setting state.

By using DownloadSettingFile() and UploadSettingFile(), you can manage the camera settings according to the shooting scene, and make it possible to restore the settings at any time. It also allows multiple cameras (same model) to share the setting status.

Before executing this API, please make sure that the media is inserted in the slot of the camera. This is due to the specifications of the camera.

After this operation, the camera reboots itself. The connection will be disconnected by restarting the camera. If CrReconnecting_OFF is specified for the fifth parameter of Connect(), execute Connect() again to establish a connection.

## Definition

```
CrError UploadSettingFile(CrDeviceHandle deviceHandle, CrUploadSettingFileType type,
CrChar* fileName);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrUploadSettingFileType | type<br><br>Specifies the type of file to upload. |
| CrChar* | fileName<br><br>Path of the file to be uploaded.<br>The extension is fixed to "DAT". |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>Other than errors above, see Status code & Error |

## Related API

- IDeviceCallback::OnWarning
- DownloadSettingFile

## Special note (details)

This API can be executed when CrDeviceProperty_CameraSetting_ReadOperationEnableStatus is Enable.

The result will be notified by OnWarning().
If the upload is successful, CrWarning_CameraSettings_Read_Result_OK will be notified by OnWarning().
If the upload fails, OnWarning() will notify you of the cause of the failure.

This API does not support all models.

You can also read(upload) the setting file by operating the camera body without using the API. In that case, the file stored in the "memory card" inserted in the media slot of the camera body will be uploaded.

For ILCE-1 : MENU > Setup > Reset/Save Settings > Save/Load Settings > Load

# MediaProfile

## GetMediaProfile

### Overview

It is an API to get the meta information of the content file recorded on the media.
In ILME-FX6, meta information such as recorded content is called "MediaProfile".

### Definition

```
CrError GetMediaProfile(CrDeviceHandle deviceHandle, CrMediaProfile slot,
CrMediaProfileInfo** mediaProfile, CrInt32u* numOfProfile);
```

### Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrMediaProfile | slot<br><br>Specifies the slot from which to get the MediaProfile.<br><br>refs. CrMediaProfile |

### Output parameters

| Type | Explanation |
|---|---|
| CrMediaProfileInfo** | mediaProfile<br><br>The CrMediaProfile list pointer. Developers should pass the address of a modifiable CrMediaProfile pointer. The value of this pointer should be initialized to nullptr.<br>This function creates a meta information list of the specified in-slot content and writes a copy to this pointer. |
| CrInt32u* | numOfProfile<br><br>An integer pointer that indicates the number of CrMediaProfileInfo returned by the mediaProfile pointer.<br><br>Developers should pass the address of a modifiable CrInt32u variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_NoApplicableInformation If there is no meta information, etc. |

## Related API

- ReleaseMediaProfile
- IDeviceCallback::OnWarning

## Special note (details)

After the movie is recorded, OnWarning() notifies CrWarning_MediaProfileChanged_Slot1 or Slot2 and notifies the media profile information update in the slot.

The content to be recorded (file format, etc.) differs depending on the camera body, so refer to the help guide for the target camera.

ReleaseMediaProfile

## Overview

This function releases the CrMediaProfileInfo allocated by GetMediaProfile().

## Definition

```
CrError ReleaseMediaProfile(CrDeviceHandle deviceHandle, CrMediaProfileInfo *
mediaProfile);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrMediaProfileInfo * | mediaProfile<br><br>CrMediaProfileInfo list pointer to release. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the Type list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API

- GetMediaProfile

## Special note (details)

None in particular

# Lens information

RequestLensInformation

## Overview

You can use this API and GetLensInformation() to get Lens information. It is valid only when a compatible lens is attached.

If you want to use the Lens information, first request the acquisition of the Lens information with this API. Then get information on GetLensInformation() after that.

- See to “How to use LensInformation” in Tips / Trouble shooting for how to use it.

## Definition

```
CrError RequestLensInformation(CrDeviceHandle deviceHandle);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>Other than errors above, see Status code & Error |

## Related API

- GetLensInformation
- IDeviceCallback::OnWarning

## Special note (details)

This API can be executed when CrDeviceProperty_LensInformationEnableStatus is Enable.

The result of this API will be notified by OnWarning(). When OnWarning notifies you of "CrWarning_RequestLensInformation_Result_Success", you can get Lens information with GetLensInformation().

If you are notified of anything other than success, it is possible that the Lens is not attached or that the Lens for which Lens information cannot be obtained is attached.

When the Lens information is updated due to Lens replacement etc., “CrWarning_LensInformationChanged” is notified by OnWarning(). If you want to use the Lens information of the replaced Lens, execute this API and GetLensInformation() to get the Lens information again.

# GetLensInformation

## Overview

It is an API to get the Lens information of the attached Lens. It can only be executed if RequestLensInformation() is successful.

- See to “How to use LensInformation” in Tips / Trouble shooting for how to use it.

## Definition

```
CrError GetLensInformation(CrDeviceHandle deviceHandle, CrLensInformation** list,
CrInt32u* numOfList);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |

## Output parameters

| Type | Explanation |
|---|---|
| CrLensInformation ** | list<br><br>The CrLensInformation list pointer. Developers should pass the address of a modifiable CrLensInformation pointer. The value of this pointer should be initialized to nullptr. Copy the Lens information stored inside the SDK-internal after the success of RequestLensInformation() to this pointer. |
| CrInt32u* | numOfList<br><br>An integer pointer that indicates the number of CrLensInformation returned by the list pointer.<br>Developers should pass the address of a modifiable CrInt32u variable. |

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None on Success<br>CrError_Api_NoApplicableInformation If numOfList is returned as zero, Make sure that Lens that can acquire Lens information is attached. If this error is returned even though the Lens for which lens information can be acquired is attached, RequestLensInformation() may not have been executed. Please do RequestLensInformation(). |

## Related API

- ReleaseLensInformation
- IDeviceCallback::OnWarning

## Special note (details)

When the Lens information is updated due to Lens replacement etc., “CrWarning_LensInformationChanged” is notified by OnWarning(). If you want to use the Lens information of the replaced Lens, execute this API and GetLensInformation() to get the Lens information again.

This API can be executed only once. If you want to get the Lens information after this API, please request again to get the Lens information from the camera with RequestLensInformation().

ReleaseLensInformation

## Overview
This function releases the CrLensInformation allocated by GetLensInformation().

## Definition

```
CrError ReleaseLensInformation(CrDeviceHandle deviceHandle, CrLensInformation* list);
```

## Input parameters

| Type | Explanation |
|---|---|
| CrDeviceHandle | deviceHandle |
| CrLensInformation* | list<br>CrLensInformation list pointer to release. |

## Output parameters

None

## Return value

| Type | Explanation |
|---|---|
| CrError | CrError_None If the Type list is released successfully<br>CrError_Init if the SDK is uninitialized<br>CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle<br>Other than errors above, see Status code & Error |

## Related API
- GetLensInformation

## Special note (details)
None in particular

# Command

## CrCommandId

Enumeration value describing command data type

Check the Function list for the Command Id(CrCommandId enumerations) supported in the current release.

See Send a Control Command and SendCommand() for usage.

Details of each Command Id are described in Parameter description.

# Device Property

## CrDeviceProperty

Class describing device properties.

Includes information about the data type, current value, and supported values. Additionally, it indicates if the property is currently modifiable.

Check the Function list for the Device Property Code(CrDevicePropertyCode enumerations) supported in the current release.

See Get the Camera Properties and SetDeviceProperty() for usage.

Details of each device properties are described in Parameter description.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| - | - | - |

### Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |
| void Alloc(const CrInt32u size, const CrInt32u getSetSize, const CrInt16u getStrSize) | It cannot be used. |
| bool IsGetEnableCurrentValue() | Checks to see if this device property is readable. |
| bool IsSetEnableCurrentValue() | Checks to see if this device property is writable. |
| CrInt32u GetCode() | Get the CrDevicePropertyCode used by this device property.<br>Defined in CrDeviceProperty.h as CrDevicePropertyCode |
| CrDataType GetValueType() | Get the CrDataType used by this device property.<br>Defined in CrDefines.h as CrDataType. |
| CrPropertyEnableFlag GetPropertyEnableFlag() | Get the CrPropertyEnableFlag that represents the enable status for this device property.<br>Defined in CrDeviceProperty.h as CrPropertyEnableFlag.<br>When enableFlag is Disable, Setter/Getter API is invalid (not guaranteed) |
| CrInt64u GetCurrentValue() | Get the current value.<br>Details of the value are given in each device property in the Parameter description. |

| | |
|---|---|
| CrInt16u* GetCurrentStr() | Get the string value when the GetValueType() is CrDataType_STR.<br>The string length is set to the first 2 bytes. The string length includes the null-terminated. |
| CrInt32u GetValueSize() | Get the total number of bytes of settable value set for values pointer. |
| CrInt8u* GetValues() | Get the pointer of settable values.<br>Details of the values are given in each device property in the Parameter description. |
| CrInt32u GetSetValueSize() | It cannot be used. Reserved function. |
| CrInt8u* GetSetValues() | It cannot be used. Reserved function. |
| void SetCode(CrInt32u codes) | Set the CrDevicePropertyCode of the device property to update.<br>Defined in CrDeviceProperty.h as CrDevicePropertyCode. |
| void SetValueType(CrDataType type) | Set the value type to update.<br>Using CrDataType defined by CrDefines.h. |
| void SetCurrentValue(CrInt64u value) | Set the value to update.<br>If CrDataType Array, only value that exist in the values pointer can be set.<br>Details of the value are given in each device property in the Parameter description. |
| void SetCurrentStr(CrInt16u* str) | Set the string value. Valid when GetValueType () is CrDataType_STR.<br>Specify the string length + 1 (for null-terminate) for the first 2 bytes. Set the string to the third byte and beyond. |

# Live View

## CrLiveViewProperty

Class for manipulating live-view properties of a device.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| - | - | - |

### Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |
| void Alloc(const CrInt32u size) | It cannot be used. |
| bool IsGetEnableCurrentValue() | Checks to see if live-view property is readable. |
| CrInt32u GetCode() | Get the CrLiveViewPropertyCode used by this live-view property. |
| CrPropertyEnableFlag GetPropertyEnableFlag() | Get the CrPropertyEnableFlag that represents the enable status for this live-view property. |
| CrFrameInfoType GetFrameInfoType() | Get the CrFrameInfoType of live-view property.<br><br>Defined in CrDeviceProperty.h as CrFrameInfoType. |
| CrInt32u GetValueSize() | Get the total number of bytes of value set for value pointer. |
| CrInt8u* GetValue() | Get the value pointer.<br><br>This pointer is set to CrFocusFrameInfo, CrMagPosInfo, CrFaceFrameInfo or CrTrackingFrameInfo. |

## Supported Live View Properties

Live View Properties supported in the current release.

| Frame name | Top row: CrLiveViewPropertyCode,<br>Bottom row: CrFrameInfoType | Class |
|---|---|---|
| AF Area | CrLiveViewProperty_AF_Area_Position | CrFocusFrameInfo |
| | CrFrameInfoType_FocusFrameInfo | |
| Focus Magnifier | CrLiveViewProperty_Focus_Magnifier_Position | CrMagPosInfo |
| | CrFrameInfoType_Magnifier_Position | |
| Face/Eye Frame | CrLiveViewProperty_FaceFrame | CrFaceFrameInfo |
| | CrFrameInfoType_FaceFrameInfo | |
| Tracking Frame | CrLiveViewProperty_TrackingFrame | CrTrackingFrameInfo |
| | CrFrameInfoType_TrackingFrameInfo | |

# CrFocusFrameInfo

Used to retrieve Focus frame info.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| type | CrFocusFrameType | The type of focus used.<br>Defined in CrDeviceProperty.h as CrFocusFrameType. |
| state | CrFocusFrameState | The state of frame.<br>Defined in CrDeviceProperty.h as CrFocusFrameState. |
| priority | CrInt8u | It cannot be used. Reserved parameter. |
| xNumerator | CrInt32u | x-axis value |
| xDenominator | CrInt32u | x-axis value |
| yNumerator | CrInt32u | y-axis value |
| yDenominator | CrInt32u | y-axis value |
| width | CrInt32u | Width of frame |
| height | CrInt32u | Height of frame |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |

## Supported Focus frame types

| Value | Explanation |
|---|---|
| CrFocusFrameType_Unknown | Undefined value |
| CrFocusFrameType_PhaseDetection_AFSensor (*) | Phase Detection AF Sensor |
| CrFocusFrameType_PhaseDetection_ImageSensor | Phase Detection Image Sensor |
| CrFocusFrameType_Wide | Wide |
| CrFocusFrameType_Zone | Zone |
| CrFocusFrameType_CentralEmphasis | Central Emphasis |
| CrFocusFrameType_ContrastFlexibleMain | Contrast Flexible Main |
| CrFocusFrameType_ContrastFlexibleAssist | Contrast Flexible Assist |
| CrFocusFrameType_Contrast | Contrast |
| CrFocusFrameType_FrameSomewhere | Frame Somewhere |

* : When the camera attached A-mount Lens & Mount Adaptor such as LA-EA4.

Supported Focus frame states

| Value | Explanation |
|---|---|
| CrFocusFrameState_Unknown | Undefined value |
| CrFocusFrameState_NotFocused | Not focused |
| CrFocusFrameState_Focused | Focused |
| CrFocusFrameState_FocusFrameSelection | Focus Frame Selection |
| CrFocusFrameState_Moving | Moving |
| CrFocusFrameState_RegistrationAF | Registration AF |
| CrFocusFrameState_Island (*) | Island |

* : When the camera attached A-mount Lens & Mount Adaptor such as LA-EA4.

# CrMagPosInfo

Used to retrieve MagnifierPosition info.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| xNumerator | CrInt32u | x-axis value |
| xDenominator | CrInt32u | x-axis value |
| yNumerator | CrInt32u | y-axis value |
| yDenominator | CrInt32u | y-axis value |
| width | CrInt32u | Width of live-view |
| height | CrInt32u | Height of live-view |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |

# CrFaceFrameInfo

Used to retrieve Face/Eye frame info. This class is also used for subject recognition.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| type | CrFaceFrameType | The type of Face/Eye frame used.<br>Defined in CrDeviceProperty.h as CrFaceFrameType. |
| state | CrFocusFrameState | The state of frame.<br>Defined in CrDeviceProperty.h as CrFocusFrameState. |
| isSelected | CrFocusFrameSelectState | It cannot be used. Reserved parameter. |
| priority | CrInt8u | It cannot be used. Reserved parameter. |
| xNumerator | CrInt32u | x-axis value |
| xDenominator | CrInt32u | x-axis value |
| yNumerator | CrInt32u | y-axis value |
| yDenominator | CrInt32u | y-axis value |
| width | CrInt32u | Width of frame |
| height | CrInt32u | Height of frame |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |

## Supported Face/Eye frame types

| Value | Explanation |
|---|---|
| CrFaceFrameType_Unknown | Undefined value |
| CrFaceFrameType_DetectedFace | Detected Face |
| CrFaceFrameType_AF_TargetFace | AF Target Face |
| CrFaceFrameType_PersonalRecognitionFace | Personal Recognition Face |
| CrFaceFrameType_SmileDetectionFace | It cannot be used. Reserved definition. |
| CrFaceFrameType_SelectedFace | Selected Face |
| CrFaceFrameType_AF_TargetSelectionFace | AF Target Selection Face |
| CrFaceFrameType_SmileDetectionSelectFace | It cannot be used. Reserved definition. |

## Supported Focus frame states

| Value | Explanation |
|---|---|
| CrFocusFrameState_NotFocused | Not focused |
| CrFocusFrameState_Focused | Focused |

CrFocusFrameState is used as in the CrFocusFrameInfo class. However, only the above is valid in this class.

# CrTrackingFrameInfo

Used to retrieve Tracking frame info.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| type | CrTrackingFrameType | The type of Tracking frame used.<br>Defined in CrDeviceProperty.h as CrTrackingFrameType. |
| state | CrFocusFrameState | The state of frame.<br>Defined in CrDeviceProperty.h as CrFocusFrameState. |
| priority | CrInt8u | It cannot be used. Reserved parameter. |
| xNumerator | CrInt32u | x-axis value |
| xDenominator | CrInt32u | x-axis value |
| yNumerator | CrInt32u | y-axis value |
| yDenominator | CrInt32u | y-axis value |
| width | CrInt32u | Width of frame |
| height | CrInt32u | Height of frame |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |

## Supported Tracking frame types

| Value | Explanation |
|---|---|
| CrTrackingFrameType_NonTargetAF | Non AF Target |
| CrTrackingFrameType_TargetAF | AF Target |

## Supported Focus frame states

| Value | Explanation |
|---|---|
| CrFocusFrameState_NotFocused | Not focused |
| CrFocusFrameState_Focused | Focused |

CrFocusFrameState is used as in the CrFocusFrameInfo class. However, only the above is valid in this class.

# CrImageInfo

Used to retrieve live-view image info. Use this class to retrieve the size of the live-view image.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| - | - | - |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| CrInt32u GetBufferSize() | Get the data size (bytes) of a live-view image. |

# CrImageDataBlock

Used for retrieving live-view image data. Allocate an object of this type to use as an output buffer.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| - | - | - |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| CrInt32u GetFrameNo() | Get the frame number. |
| void SetSize(CrInt32u size) | Set the maximum size(bytes) that can save live-view images. Use the size(bytes) obtained by CrImageInfo::GetBufferSize() |
| CrInt32u GetSize() | Get the size set in SetSize(). |
| void SetData(CrInt8u* data) | Set the receive pointer for live-view image. |
| CrInt32u GetImageSize() | Get the live-view image(jpeg) data size. |
| CrInt8u* GetImageData() | Get the pointer of live-view image(jpeg) data. |

# Contents Transfer

## CrMtpFolderInfo

Class describing content storage folder.

Has a folder handle and date information. This folder handle is used to get the "CrContentHandle" needed to pull out the content.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| handle | CrFolderHandle | Date folder handle. |
| folderNameSize | CrInt32u | Size of the folderName. |
| folderName | CrChar* | Folder name. format : “YYYY-MM-DD” |

### Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |
| void Alloc(const CrInt32u size) | It cannot be used. |

# CrMtpContentsInfo

Class describing content.

Includes information about the content file name, content file size, and supported values. This information is used to pull content from the media inserted in the camera slot.

## Member Variables

| Name | Type | Summary |
|---|---|---|
| handle | CrContentHandle | Content handle. |
| parentFolderHandle | CrFolderHandle | Handle of the Date Folder where the content is saved. |
| contentSize | CrInt64u | Size of the content. |
| dateChar | CrChar[16] | Shooting date and time.<br>format : “YYYYMMDDThhmmss”<br>ex) 7/16/2010 1:25:46 PM= 20100716T132546 |
| width | CrInt32u | Content width. unit : pixel |
| height | CrInt32u | Content height. unit : pixel |
| fileNameSize | CrInt32u | Size of the fileName. |
| fileName | CrChar* | Content name.<br>Note :<br>The AVCHD file name is in "YYYYMMDDhhmmss" format (datetime).<br>ex) 20100716132546.MTS |

## Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |
| void Alloc(const CrInt32u size) | It cannot be used. |

# Display string

## CrDisplayStringListInfo

Class describing display information.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| dataType | CrDataType | Type of value |
| listType | CrDisplayStringType | Type of display string |
| value | CrInt64u | Value that means a display string |
| displayStringSize | CrInt32u | Length of display string |
| displayString | CrInt8u* | Display string |

### Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |
| void Alloc(const CrInt32u size) | It cannot be used. |

# CrDisplayStringType

Enumerate the kind of list-type.

## Member Enumerations

| Definition | Summary | |
|---|---|---|
| | ILME-FX6 | Other Models |
| CrDisplayStringType_AllList | Request all list types | |
| CrDisplayStringType_BaseLook_AELevelOffset_ExposureValue | Rightmost column of Menu > Paint/Look > Base Look > Select | - |
| CrDisplayStringType_BaseLook_Input_Display | Input column of Menu > Paint/Look > Base Look > Select | - |
| CrDisplayStringType_BaseLook_Name_Display | Base Look Name column of Menu > Paint/Look > Base Look > Select | Table of Menu > Exposure/Color > Color/Tone > Select LUT |
| CrDisplayStringType_BaseLook_Output_Display | Output column of Menu > Paint/Look > Base Look > Select | - |
| CrDisplayStringType_SceneFile_Name_Display | Base Look Name column of Menu > Paint/Look > Scene File > Recall when Menu > Project > Base Setting > Shooting Mode is “Custom” | - |
| CrDisplayStringType_ShootingMode_Cinema_ColorGamut_Display | Menu > Project > Cine EI Setting > Color Gamut | - |
| CrDisplayStringType_ShootingMode_TargetDisplay_Display | Menu > Project > Base Setting > Target Display | - |
| CrDisplayStringType_Camera_Gain_BaseISO_Display | Menu > Shooting > ISO/Gain/EI > BaseISO | Menu > Exposure/Color > Exposure > Base ISO |
| CrDisplayStringType_Video_EIGain_Display | Menu > Shooting > ISO/Gain/EI > Exposure Index<H>, <M>, <L> when Menu > Project > Base Setting > Shooting Mode is “Cine EI” | Menu > Exposure/Color > Exposure > Exposure Index |

| | | |
|---|---|---|
| CrDisplayStringType_Button_Assign_Display | Menu > Project > Assignable Button | - |
| CrDisplayStringType_Button_Assign_ShortDisplay | Abbreviation string for Menu > Project > Assignable Button. This menu is not in the camera body. | - |
| CrDisplayStringType_CreativeLook_Name_Display | - | Menu > Exposure/Color > Color/Tone > Creative Look > Look |

# MediaProfile

## CrMediaProfileInfo

Class describing display information.

For the content type and extension, refer to the help guide of the main unit because it is the main unit specification.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| contentName | CrInt8u* | Name of content |
| contentUrl | CrInt8u* | Url of content |
| contentType | CrInt8u* | Type of content |
| contentFrameRate | CrInt8u* | Frame rate of content |
| contentAspectRatio | CrInt8u* | Aspect ratio of content |
| contentChannel | CrInt8u* | Channel of content |
| contentVideoType | CrInt8u* | Video type of content |
| contentAudioType | CrInt8u* | Audio type of content |
| proxyUrl | CrInt8u* | Url of proxy content |
| proxyType | CrInt8u* | Type of proxy content |
| proxyFrameRate | CrInt8u* | Frame rate of proxy content |
| proxyAspectRatio | CrInt8u* | Aspect ratio of proxy content |
| proxyChannel | CrInt8u* | Channel of proxy content |
| proxyVideoType | CrInt8u* | Video type of proxy content |
| proxyAudioType | CrInt8u* | Audio type of proxy content |
| thumbnailUrl | CrInt8u* | Url of thumbnail image file |

Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |

## CrMediaProfile

Enumerate of MediaProfile slot.

Member Enumerations

| Definition | Summary |
|---|---|
| CrMediaProfile_Slot1 | Media such as SD card inserted in slot 1 |
| CrMediaProfile_Slot2 | Media such as SD card inserted in slot 2 |

# Lens Information

## CrLensInformation

Class describing display information.

### Member Variables

| Name | Type | Summary |
|---|---|---|
| type | CrLensInformationType | Type of LensInformation |
| dataVersion | CrInt16u | Data Version(100 -fold value) |
| normalizedValue | CrInt32u | Normalized focus position value |
| focusPosition | CrInt32u | Focus position<br>ex)20 = 0.2feet/meter |

### Member Functions

| Signature | Description |
|---|---|
| Constructor | - |
| Destructor | - |
| Copy Constructor | - |

# CrLensInformationType

Enumerate the kind of list-type.

## Member Enumerations

| Definition | Summary |
|---|---|
| CrLensInformationType_Undefined | Unavailable |
| CrLensInformationType_Feet | Focus position information whose unit is Feet |
| CrLensInformationType_Meter | Focus position information whose unit is Meter |

# Callback Interface

## IDeviceCallback

The callback interface of the SDK. This interface is used by the Camera Remote SDK to communicate the result of various asynchronous events to the user.

The user must implement a class deriving from this interface to use the SDK. This derived class should be passed to the Connect API to establish the callback communication channel with the SDK.

### Pure Virtual Functions

| Signature | Description |
|---|---|
| virtual void OnConnected(DeviceConnectionVersion version) | Called by the SDK when a device is successfully connected |
| virtual void OnDisconnected(CrInt32u error) | Called by the SDK when a device disconnects. The error code may indicate a reason |
| virtual void OnPropertyChanged() | Called by the SDK when a device property changes |
| virtual void OnLvPropertyChanged() | Called by the SDK when a LiveView property changes |
| virtual void OnCompleteDownload(CrChar *filename, CrInt32u type = 0xFFFFFFFF) | Called by the SDK when a capture image or setting file has completely been transferred to the host device.<br><br>When capture image transfer is complete, the type parameter is 0xFFFFFFFF.<br><br>When DownloadSettingFile() succeeds, type parameter becomes CrDownloadSettingFileType_Setup. |
| virtual void OnWarning(CrInt32u warning) | Called when the SDK detects a warning. The warning code is passed back to the application as a parameter |
| virtual void OnError(CrInt32u error) | Called when the SDK detects an error. The error code is passed back to the application as a parameter |
| virtual void OnPropertyChangedCodes(CrInt32u num, CrInt32u* codes) | Called by the SDK when a device property changes.<br><br>The difference from OnPropertyChanged() is that you can get the updated device property code list. If you pass the device property code list received by this callback to GetSelectDeviceProperties(), you can get only the updated property information. Performance improvement can be expected by minimizing the amount of receive data. |

| | |
|---|---|
| virtual void OnLvPropertyChangedCodes(CrInt32u num, CrInt32u* codes) | Called by the SDK when a LiveView property changes |
| virtual void OnNotifyContentsTransfer(CrInt32u notify, CrContentHandle handle, CrChar* filename = 0) | Called when content transfer starts and ends, or when transfer fails.<br><br>The filename parameter is the name (including path) of the content that will be set when the content transfer is complete.<br>The filename parameter is not set when content transfer is started or when content transfer fails. |

# ICrCameraObjectInfo

Your application can access to the specified camera information that is enumerated by EnumCameraObjects() using this interface.

The information retrieved from this interface is useful for displaying various information about the corresponding device to the end user of an application utilising the Camera Remote SDK. The information provided by this class is also required when establishing a new connection to a camera device. It should be provided when calling the Connect API.

The user should never manually free these objects by calling free or delete. Instead, the user should call ICrCameraObjectInfo::Release. This passes responsibility for releasing the allocated memory to the SDK, where it can be properly released.

## Pure Virtual Functions

| Signature | Description |
|---|---|
| virtual void Release() | Calls the SDK to destroy the allocated object |
| virtual CrChar* GetName() const | Gets the friendly device name as a null-terminated character string<br><br>(Friendly device name is not available through SDK, currently.) |
| virtual CrInt32u GetNameSize() const | Gets the size of the name string |
| virtual CrChar* GetModel() const | Gets the device model name as a null-terminated character string |
| virtual CrInt32u GetModelSize() const | Gets the size of the model string |
| virtual CrInt16 GetUsbPid() const | Gets the product id of a USB device |
| virtual CrInt8u* GetId() const | Gets the pointer to the device id data buffer |
| virtual CrInt32u GetIdSize() const | Gets the id data size |
| virtual CrInt32u GetIdType() const | Gets the id data type (binary or string data) |
| virtual CrInt32u GetConnectionStatus() const | Gets the current connection status of the device |
| virtual CrChar* GetConnectionTypeName() const | Gets the connection type string |
| virtual CrChar* GetAdaptorName() const | Gets the adaptor name string |
| virtual CrChar* GetGuid() const | It cannot be used. Reserved function. |
| virtual CrChar* GetPairingNecessity() const | Gets the need for pairing |
| virtual CrInt16u GetAuthenticationState() const | It cannot be used. Reserved function. |
| virtual CrInt32u GetSSHsupport() const | Gets the device SSH Support |

# ICrEnumCameraObjectInfo

The virtual interface for interacting with enumerated device info list created by the SDK.

This is the enumerator object interface to access the list of connectable cameras. Your application can get the access interface to the each camera using GetCameraObjectInfo().

A “connectable” device fulfils three requirements. One, the device itself supports PC Remote Control features. Two, the device model is supported by the current Camera Remote SDK release. Three, the connection method used by the device is supported by the current Camera Remote SDK. All three requirements must be fulfilled for the device information to be populated in the list.

All ICrEnumCameraObjectInfo interface objects are allocated internally by the SDK before having their address passed back to the user. The user should never manually free these objects by calling free or delete. Instead, the user should call ICrEnumCameraObjectInfo::Release. This passes responsibility for releasing the allocated memory to the SDK, where it can be properly released.

## Pure Virtual Functions

| Signature | Description |
|---|---|
| virtual void Release() | Calls the SDK to destroy the allocated device info list |
| virtual CrInt32u GetCount() const | Returns the number of device info objects in the allocated list |
| virtual const ICrCameraObjectInfo* GetCameraObjectInfo(CrInt32u index) const | Get a pointer to the ICrCameraObjectInfo at the index specified |

# Status code & Error

Major status codes are below. The "error" member is defined as [error_code, error_message].
The error_message may vary depending on the camera models.

## Error Category

| Name | Summary |
|---|---|
| CrError_None | No error |
| CrError_Generic | Uncategorized errors |
| CrError_File | File errors |
| CrError_Connect | Communication errors |
| CrError_Memory | Memory errors |
| CrError_Api | API errors |
| CrError_Init | Initialization errors |
| CrError_Polling | Polling errors |
| CrError_Adaptor | Adapter errors |
| CrError_Device | Device errors |
| CrError_Contents | Content transfer errors |

## CrError_None

## CrError_Generic

| Name | Summary |
|---|---|
| CrError_Generic_Unknown | Uncategorized errors |
| CrError_Generic_Notimpl | Not implemented |
| CrError_Generic_Abort | Processing was aborted |
| CrError_Generic_NotSupported | Not supported |
| CrError_Generic_SeriousErrorNotSupported | Not supported |
| CrError_Generic_InvalidHandle | Not valid handle |
| CrError_Generic_InvalidParameter | Invalid parameter |

## CrError_File

| Name | Summary |
|---|---|
| CrError_File_Unknown | Unknown file errors |
| CrError_File_IllegalOperation | Illegal operation (e.g., loading without opening) |
| CrError_File_IllegalParameter | Illegal parameter |
| CrError_File_EOF | EOF |
| CrError_File_OutOfRange | Operation, such as seek, is out of range |
| CrError_File_NotFound | File not found |
| CrError_File_DirNotFound | Directory not found |
| CrError_File_AlreadyOpened | Already opened |
| CrError_File_PermissionDenied | No access permission |
| CrError_File_StorageFull | Host storage is full |
| CrError_File_AlreadyExists | Already exists |
| CrError_File_TooManyOpenedFiles | Too many open files |
| CrError_File_ReadOnly | Read-Only file |
| CrError_File_CantOpen | Cannot open |
| CrError_File_CantClose | Cannot close |
| CrError_File_CantDelete | Cannot delete |
| CrError_File_CantRead | Cannot read |
| CrError_File_CantWrite | Cannot write |
| CrError_File_CantCreateDir | Cannot create a directory |
| CrError_File_OperationAbortedByUser | Processing was aborted by user |
| CrError_File_UnsupportedOperation | API not supported for the platform was called |
| CrError_File_NotYetCompleted | Operation is not completed |
| CrError_File_Invalid | The file is no longer valid because the volume for the file was altered |
| CrError_File_StorageNotExist | The specified network resource or device is no longer available |
| CrError_File_SharingViolation | Sharing violation |
| CrError_File_Rotation | Invalid file orientation |
| CrError_File_SameNameFull | Too many same-name files |

## CrError_Connect

| Name | Summary |
|---|---|
| CrError_Connect_Unknown | Other errors classified as connection except below |
| CrError_Connect_Connect | A connection request failed through the USB |
| CrError_Connect_Release | Release failed |
| CrError_Connect_GetProperty | Getting property failed |
| CrError_Connect_SendCommand | Sending command failed |
| CrError_Connect_HandlePlugin | Illegal handle plug-in |
| CrError_Connect_Disconnected | A connection disconnected |
| CrError_Connect_TimeOut | A connection operation timed out |
| CrError_Reconnect_TimeOut | Reconnection operations timed out. |
| CrError_Connect_FailRejected | Connection rejected and failed |
| CrError_Connect_FailBusy | Connection failed due to processing in progress |
| CrError_Connect_FailUnspecified | Unspecified connection failure |
| CrError_Connect_Cancel | Connection canceled |
| CrError_Connect_SessionAlreadyOpened | Session is open |
| CrError_Connect_ContentsTransfer_NotSupported | Connection to the content transfer mode on a non-supporting model. |
| CrError_Connect_SSH_NotSupported | Cameras that do not support SSH authentication |
| CrError_Connect_SSH_InvalidParameter | Illegal parameter |
| CrError_Connect_SSH_ServerConnectFailed | Cannot connect to SSH server |
| CrError_Connect_SSH_ServerAuthenticationFailed | SSH authentication failed (fingerprint difference) |
| CrError_Connect_SSH_UserAuthenticationFailed | SSH authentication failed (User name or Password incorrect) |
| CrError_Connect_SSH_PortForwardFailed | Port forwarding failure (the specified port number cannot be used, etc.) |
| CrError_Connect_SSH_GetFingerprintFailed | Fingerprint data acquisition failure |

## CrError_Memory

| Name | Summary |
|---|---|
| CrError_Memory_Unknown | Unknown memory error |
| CrError_Memory_OutOfMemory | Cannot allocate memory |
| CrError_Memory_InvalidPointer | Invalid pointer |
| CrError_Memory_Insufficient | Allocate memory insufficient |

## CrError_Api

| Name | Summary |
|---|---|
| CrError_Api_Unknown | Unknown API error |
| CrError_Api_Insufficient | Incorrect parameter |
| CrError_Api_InvalidCalled | Invalid API call |
| CrError_Api_NoApplicableInformation | No applicable information exists. |
| CrError_Api_OutOfModelList | Outside the scope of the camera model list |
| CrError_Api_NotSupportModelOfUSB | Model that does not support USB connection |
| CrError_Api_NotSupportModelOfEthernet | Model that does not support Ethernet connection |
| CrError_Api_InvalidSerialNumber | Invalid serial number |
| CrError_Api_InvalidIpAddress | Invalid serial IP Address |
| CrError_Api_InvalidMacAddress | Invalid serial Mac Address |

## CrError_Init

## CrError_Polling

| Name | Summary |
|---|---|
| CrError_Polling_Unknown | Unknown polling error |
| CrError_Polling_InvalidVal_Intervals | Invalid polling interval setting value |

## CrError_Adaptor

| Name | Summary |
|---|---|
| CrError_Adaptor_Unknown | Unknown adapter error |
| CrError_Adaptor_InvalidProperty | A property that doesn't exist was used |
| CrError_Adaptor_GetInfo | Getting information failed |
| CrError_Adaptor_Create | Creation failed |
| CrError_Adaptor_SendCommand | Sending command failed |
| CrError_Adaptor_HandlePlugin | Illegal handle plug-in |
| CrError_Adaptor_CreateDevice | Device creation failed |
| CrError_Adaptor_EnumDevice | Enumeration of device information failed |
| CrError_Adaptor_Reset | Reset failed |

| | |
|---|---|
| CrError_Adaptor_Read | Read failed |
| CrError_Adaptor_Phase | Parse failed |
| CrError_Adaptor_DataToWiaItem | Failed to set data as WIA item |
| CrError_Adaptor_DeviceBusy | The setting side is busy |
| CrError_Adaptor_Escape | Escape failed |

## CrError_Device

| Name | Summary |
|---|---|
| CrError_Device_Unknown | Unknown device error |

## CrError_Contents

| Name | Summary |
|---|---|
| CrError_Contents_Unknown | Unknown Contents error |
| CrError_Contents_InvalidHandle | The specified handle is invalid |
| CrError_Contents_DateFolderList_NotRetrieved | Before getting date folder List |
| CrError_Contents_ContentsList_NotRetrieved | Before getting content handles array |
| CrError_Contents_Transfer_Unsuccess | Content transfer failed |
| CrError_Contents_Transfer_Cancel | Content transfer canceled |
| CrError_Contents_RejectRequest | Rejected request |

## CrWarning

| Name | Summary |
|---|---|
| CrWarning_Unknown | Warning: unknown warning |
| CrWarning_Connect_Reconnected | Warning: reconnected |
| CrWarning_Connect_Reconnecting | Warning: reconnecting |
| CrWarning_Connect_Already | Warning: already connected |
| CrWarning_Connect_OverLimitOfDevice | Warning: connection limitations<br>Exceeded the number of connectable devices |
| CrWarning_File_StorageFull | Warning: host storage is almost full<br>If you need to check camera storage, please use Device Property “Media SLOTx Remaining number shots”. |
| CrWarning_SetFileName_Failed | Warning: file name setting error |
| CrWarning_GetImage_Failed | Warning: error in getting image |
| CrWarning_FailedToSetCWB | Not notified. Reserved definition. |
| CrWarning_NetworkErrorOccurred | Warning: network error occurred |
| CrWarning_NetworkErrorRecovered | Warning: recovered from network error |
| CrWarning_Format_Failed | Warning: formatting failed |
| CrWarning_Format_Invalid | Warning: invalid formatting |

| | |
|---|---|
| CrWarning_Format_Complete | Warning: formatting complete |
| CrWarning_Format_Canceled | Warning: formatting canceled |
| CrWarning_DateTime_Setting_Result_Invalid | Warning: invalid setting |
| CrWarning_DateTime_Setting_Result_OK | Warning: DateTime setting succeeded |
| CrWarning_DateTime_Setting_Result_Parameter_Error | Warning: DateTime setting failed (Parameter Error) |
| CrWarning_DateTime_Setting_Result_Exclusion_Error | Warning: DateTime setting failed (Exclusion Error) |
| CrWarning_DateTime_Setting_Result_System_Error | Warning: DateTime setting failed (System Error) |
| CrWarning_Frame_NotUpdated | Warning: live view frame not updated |
| CrWarning_ZoomAndFocusPosition_Invalid | Warning: zoom & focus position preset |
| CrWarning_ZoomAndFocusPosition_DifferentLens | Warning: lens at save and the attached lens are different |
| CrWarning_ZoomAndFocusPosition_InvalidLens | Warning: invalid lens is attached |
| CrWarning_ContentsTransferMode_Invalid | Warning: Camera cannot be in content transfer mode |
| CrWarning_ContentsTransferMode_DeviceBusy | Warning: Camera cannot be in content transfer mode (DeviceBusy) |
| CrWarning_ContentsTransferMode_StatusError | Warning: Camera cannot be in content transfer mode (StatusError) |
| CrWarning_ContentsTransferMode_CanceledFromCamera | Warning: Canceled on the LCD screen of the camera body |
| CrWarning_ContentsTransferCancel_Success | Warning: Successful cancellation of content transfer |
| CrWarning_ContentsTransferCancel_Error | Warning: Failed to cancel content transfer |
| CrWarning_CameraSettings_Read_Result_Invalid | Warning: Invalid setting file |
| CrWarning_CameraSettings_Read_Result_OK | Warning: Successful upload of setting file |
| CrWarning_CameraSettings_Read_Result_NG | Warning: Failed to update the setting file |
| CrWarning_CameraSettings_Save_Result_NG | Warning: Failed to download the setting file |
| CrWarning_RequestDisplayStringList_Success | Warning: Successful get DisplayStringList |
| CrWarning_RequestDisplayStringList_Error | Warning: Failed to get DisplayStringList |
| CrWarning_DisplayListChanged_BaseLook_AELevelOffsetExposureValueList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_BaseLook_InputDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_BaseLook_NameDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_BaseLook_OutputDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_SceneFile_NameDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_ShootingMode_CinemaColorGamutDisplayList | Not notified. Reserved definition. |

| | |
|---|---|
| CrWarning_DisplayListChanged_ShootingMode_TargetDisplayDisplayList | Not notified. Reserved definition. |
| CrWarning_DisplayListChanged_Camera_GainBaseISODisplayList | Not notified. Reserved definition. |
| CrWarning_DisplayListChanged_Video_EIGainDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_Button_AssignDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_DisplayListChanged_Button_AssignShortDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_MediaProfileChanged_Slot1 | Warning: MediaProfile update for media inserted in slot1 |
| CrWarning_MediaProfileChanged_Slot2 | Warning: MediaProfile update for media inserted in slot2 |
| CrWarning_LensInformationChanged | Warning: Lens Information update |
| CrWarning_RequestLensInformation_Result_Success | Warning: Successful get of Lens information |
| CrWarning_RequestLensInformation_Result_DeviceBusy | Warning: Failed to get Lens information(Device Busy) |
| CrWarning_RequestLensInformation_Result_Error | Warning: Failed to get Lens information(Other than Device Busy) |
| CrWarning_DisplayListChanged_CreativeLook_NameDisplayList | Warning: Display strings updated. Please refer to CrDisplayStringType |
| CrWarning_CustomWBCapture_Result_Invalid | Not notified. Reserved definition. |
| CrWarning_CustomWBCapture_Result_OK | Warning: Successful CustomWB Capture. |
| CrWarning_CustomWBCapture_Result_NG | Warning: Failed to CustomWB Capture. |

## CrNotify

| Name | Summary |
|---|---|
| CrNotify_All_Download_Complete | Notification: download completed |
| CrNotify_Captured_Event | Notification: Still image capture complete.<br>Not supporting Products : ILCE-9M2, ILCE-7RM4, ILME-FX6 |
| CrNotify_ContentsTransfer_Start | Notification: Content transfer started |
| CrNotify_ContentsTransfer_Complete | Notification: Content transfer completed |

Please ignore Error/Warning/Notify except above.

# Parameter description

## CrCommandId_Release

Release the shutter to shoot

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Up the shutter button |
| CrCommandParam_Down | Down the shutter button<br>After executing “Down”, send “Up” to cancel the Down status. |

## CrCommandId_MovieRecord

Control Movie Rec button

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Specify “Up” when stop movie recording |
| CrCommandParam_Down | Specify “Down” when you start movie recording<br><br>Note:<br>After starting movie recording, please check the movie recording status with CrDeviceProperty_RecordingState.<br><br>Caution:<br>The below models can be start or stop with the “Down”, but please execute “Up” after “Down” at once.<br>ILCE-1, ILCE-9M2, ILCE-7RM4A, ILCE-7RM4, ILCE-7SM3, ILCE-7C and DSC-RX0M2. |

## CrCommandId_MediaFormat

Formatting the media. refs Select Media Format.

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Specify when initializing the media in SLOT1<br>Ex. “CrCommandId_MediaFormat” with “CrCommandParam_Up” |
| CrCommandParam_Down | Specify when initializing the media in SLOT2<br>Ex. “CrCommandId_MediaFormat” with “CrCommandParam_Down” |

## CrCommandId_MediaQuickFormat

Quick formatting the media

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Specify when quick and simple initializing the media in SLOT1<br>Ex. “CrCommandId_MediaQuickFormat” with “CrCommandParam_Up” |
| CrCommandParam_Down | Specify when quick and simple initializing the media in SLOT2<br>Ex. “CrCommandId_MediaQuickFormat” with “CrCommandParam_Down” |

## CrCommandId_CancelMediaFormat

Cancel the media format

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Release the down state of the Cancel button |
| CrCommandParam_Down | Press the Cancel button of the media format.<br>After executing Down, please release the Down state by executing Up.<br>When CrDeviceProperty_Cancel_Media_FormatEnableStatus is Enable, it is possible to cancel Full format(CrCommandId_MediaFormat) by sending this command.<br>However, once you start Full format, you will not be able to access the image data in the media even if you perform this cancel operation. (The media will be the same state as after Quick format is executed. |

## CrCommandId_S1andRelease

Shutter Half Release and Release to shoot.

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Up the shutter button |
| CrCommandParam_Down | Down the shutter button<br>After executing “Down”, send “Up” to cancel the Down status. |

## CrCommandId_CancelContentsTransfer

Cancel content transfer

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify when canceling the content transfer process<br>Check the CrDeviceProperty_ContentsTransferCancelEnableStatus status to see if you can cancel or not. |

## CrCommandId_CameraSettingsReset

Initialize the settings of the camera body

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Down | Press the setting reset button on the camera body.<br>Valid when CrDeviceProperty_CameraSettingsResetEnableStatus is Enable. This operation resets the camera settings and restarts the camera.<br><br>Caution:<br>The connection will be disconnected by restarting the camera. If CrReconnecting_OFF is specified for the fifth parameter of Connect(), execute Connect() again to establish a connection. |

## CrCommandId_APS_C_or_Full_Switching

Switch the image sensor to APS-C or Full.

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify “Down” to switch between APS-C and Full.<br><br>Valid when CrDeviceProperty_APS_C_or_Full_SwitchingEnableStatus is Enable. Each time you execute a command, the image sensor of the camera switches between APS-C size and full size. You can check the current value with CrDeviceProperty_APS_C_or_Full_SwitchingSetting. |

## CrCommandId_MovieRecButtonToggle

Control Movie Rec Button (2nd).

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify “Down” when you start movie recording and stop movie recording.<br>Valid when CrDeviceProperty_MovieRecButtonToggleEnableStatus is Enable.<br><br>Note:<br>After starting movie recording, please check the movie recording status with CrDeviceProperty_RecorderMainStatus. |

## CrCommandId_CancelRemoteTouchOperation

Cancel Remote Touch Operation

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify when canceling the Remote Touch Operation<br>Check the CrDeviceProperty_CancelRemoteTouchOperationEnableStatus status to see if you can cancel or not. |

## CrCommandId_PixelMapping

Execute Pixel Mapping

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Down | Specify "Down" when you want to image sensor optimization(Pixel Mapping).<br>Valid when CrDeviceProperty_PixelMappingEnableStatus is Enable. When the Pixel Mapping is finished, the camera restart.<br><br>Note:<br>"Auto Pixel Mapping" is not performed with this function. It is recommended to perform CrCommandId_PixelMapping periodically. |

## CrCommandId_TimeCodePresetReset

Execute Time Code Preset Reset

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify "Down" when you want to reset Time Code Preset.<br>Valid when CrDeviceProperty_TimeCodePresetResetEnableStatus is Enable. |

## CrCommandId_UserBitPresetReset

Execute User Bit Preset Reset

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify "Down" when you want to reset User Bit Preset.<br>Valid when CrDeviceProperty_UserBitPresetResetEnableStatus is Enable. |

## CrCommandId_SensorCleaning

Execute Sensor Cleaning

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Down | Specify "Down" when you want to Sensor Cleaning.<br>Valid when CrDeviceProperty_SensorCleaningEnableStatus is Enable.<br>When the Sensor Cleaning is finished, the camera restart. |

## CrCommandId_PictureProfileReset

Execute Picture Profile Reset

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify "Down" when you want to reset Picture Profile.<br>Valid when CrDeviceProperty_PictureProfileResetEnableStatus is Enable. |

## CrCommandId_CreativeLookReset

Execute Creative Look Reset

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Up | Be sure to specify "Up" after specifying "Down". |
| CrCommandParam_Down | Specify "Down" when you want to reset Creative Look.<br>Valid when CrDeviceProperty_CreativeLookResetEnableStatus is Enable. |

## CrCommandId_PowerOff

Execute Power Off

| Parameter Code | Explanation |
|---|---|
| CrCommandParam_Down | Specify "Down" to turn off the power.<br><br>Note:<br>To turn on the power after the power is turned off, please operate the camera's ON/OFF (Power) switch or input the shutter trigger signal via USB.<br><br>The camera will not start up even if the main power of the camera is turned on after the camera is turned off by this command while power is supplied via USB. In this case, please input the shutter trigger signal via USB to start the camera.<br><br>"Auto Pixel Mapping" is not performed with this function. It is recommended to perform CrCommandId_PixelMapping periodically. |

## CrDeviceProperty_S1

Get/Set the Shutter button half release

| Parameter Code | Explanation |
|---|---|
| CrLockIndicator_Unlocked | Unlock |
| CrLockIndicator_Locked | Lock |

## CrDeviceProperty_AEL

Get the AELock Indication and control AEL button

| Parameter Code | Explanation |
|---|---|
| CrLockIndicator_Unlocked | Unlock |
| CrLockIndicator_Locked | Lock |

## CrDeviceProperty_FEL

Get the FEL Lock Indication and control FEL button

| Parameter Code | Explanation |
|---|---|
| CrLockIndicator_Unlocked | Unlock |
| CrLockIndicator_Locked | Lock |

## CrDeviceProperty_AWBL

Get the AWBLock Indication and control AWBL button

| Parameter Code | Explanation |
|---|---|
| CrLockIndicator_Unlocked | Unlock |
| CrLockIndicator_Locked | Lock |

## CrDeviceProperty_FNumber

Get/Set the Aperture Value (F-Number)

| Value | Explanation |
|---|---|
| CrFnumber_Nothing | Nothing to display |
| CrFnumber_Unknown | Display "--" |
| Other than above values | The value is obtained by multiplying a real FNumber value by 100.<br><br>e.g.) 0x0190 =400 (means F-4)<br><br>0x03B6 = 950 (means F-9.5) |

## CrDeviceProperty_ExposureBiasCompensation

Get/Set the Exposure Bias Compensation

| Value | Explanation |
|---|---|
| - | The value is obtained by multiplying a real Exposure Bias Compensation value by 1000.<br><br>e.g.) 0xEC78 = -5000 (means -5.0Ev)<br>0x0000 = 0 (means 0.0Ev)<br>0x1388 = 5000 (means 5.0Ev) |

## CrDeviceProperty_FlashCompensation

Get/Set the Flash Compensation

| Value | Explanation |
|---|---|
| - | The value is obtained by multiplying a real Flash Compensation value by 1000.<br><br>e.g.) 0xEC78 = -5000 (means -5.0Ev)<br>0x0000 = 0 (means 0.0Ev)<br>0x1388 = 5000 (means 5.0Ev) |

## CrDeviceProperty_ShutterSpeed

Get/Set the Shutter Speed

| Value | Explanation |
|---|---|
| CrShutterSpeed_Bulb | BULB |
| CrShutterSpeed_Nothing | nothing to display |
| Other than above values | The real value of shutter speed (Upper two bytes: numerator, Lower two bytes: denominator)<br><br>In the case of the shutter speed is displayed as "Real Number" on the camera, the denominator is fixed 0x000A.<br><br>e.g.) 0x000F000A: 0x000F (means 15) / 0x0000A (means 10) = 1.5"<br><br>In the case of the shutter speed is displayed as "Fraction Number" on the camera, the numerator is fixed 0x0001.<br><br>e.g.) 0x000103E8: 0x0001 (means 1) / 0x03E8 (means 1000) = 1/1000 |

## CrDeviceProperty_IsoSensitivity

Get/Set the ISO Sensitivity

| Value | Explanation |
|---|---|
| - | value : bit 28-31 extension, bit 24-27 ISO mode , bit 0-23 ISO value.<br><br>Real ISO value : when bits 0-23 are other than CrISO_AUTO(0xFFFFFF).<br><br>e.g.) 0x00000140 = 320 |

## CrDeviceProperty_FocusArea

Get/Set the Focus Area

| Parameter Code | Explanation |
|---|---|
| CrFocusArea_Wide | Wide |
| CrFocusArea_Zone | Zone |
| CrFocusArea_Center | Center |
| CrFocusArea_Flexible_Spot_S | Flexible spot S |
| CrFocusArea_Flexible_Spot_M | Flexible spot M |
| CrFocusArea_Flexible_Spot_L | Flexible spot L |
| CrFocusArea_Expand_Flexible_Spot | Expand flexible spot |
| CrFocusArea_Flexible_Spot | Flexible spot |
| CrFocusArea_Tracking_Wide | Tracking on AF wide |
| CrFocusArea_Tracking_Zone | Tracking on AF zone |
| CrFocusArea_Tracking_Center | Tracking on AF center |
| CrFocusArea_Tracking_Flexible_Spot_S | Tracking on AF flexible spot S |
| CrFocusArea_Tracking_Flexible_Spot_M | Tracking on AF flexible spot M |
| CrFocusArea_Tracking_Flexible_Spot_L | Tracking on AF flexible spot L |
| CrFocusArea_Tracking_Expand_Flexible_Spot | Tracking on expand flexible spot |
| CrFocusArea_Tracking_Flexible_Spot | Tracking on AF flexible spot |

## CrDeviceProperty_ExposureProgramMode

Get/Set the Exposure Program Mode

| Parameter Code | Explanation |
|---|---|
| CrExposure_M_Manual | Manual(M) |
| CrExposure_P_Auto | Automatic(P) |
| CrExposure_A_AperturePriority | Aperture Priority(A) |
| CrExposure_S_ShutterSpeedPriority | Shutter Priority(S) |
| CrExposure_Program_Creative | Program Creative(greater depth of field) |
| CrExposure_Program_Action | Program Action(faster shutter speed) |
| CrExposure_Portrait | Portrait |
| CrExposure_Auto | Auto |
| CrExposure_Auto_Plus | Auto+ |
| CrExposure_P_A | P_A |
| CrExposure_P_S | P_S |
| CrExposure_Sports_Action | Sports Action |
| CrExposure_Sunset | Sunset |
| CrExposure_Night | Night Scene |

| | |
|---|---|
| CrExposure_Landscape | Landscape |
| CrExposure_Macro | Macro |
| CrExposure_HandheldTwilight | Hand-held Twilight |
| CrExposure_NightPortrait | Night Portrait |
| CrExposure_AntiMotionBlur | Anti Motion Blur |
| CrExposure_Pet | Pet |
| CrExposure_Gourmet | Gourmet |
| CrExposure_Fireworks | Fireworks |
| CrExposure_HighSensitivity | High Sensitivity |
| CrExposure_MemoryRecall | MemoryRecall(MR) |
| CrExposure_ContinuousPriority_AE_8pics | Tele-Zoom Continuous Priority AE 8pics |
| CrExposure_ContinuousPriority_AE_10pics | Tele-Zoom Continuous Priority AE 10pics |
| CrExposure_ContinuousPriority_AE_12pics | Continuous Priority AE12pics |
| CrExposure_3D_SweepPanorama | 3D Sweep Panorama Shooting |
| CrExposure_SweepPanorama | Sweep Panorama Shooting |
| CrExposure_Movie_P | Movie Recording(P) |
| CrExposure_Movie_A | Movie Recording(A) |
| CrExposure_Movie_S | Movie Recording(S) |
| CrExposure_Movie_M | Movie Recording(M) |
| CrExposure_Movie_Auto | Movie Recording(AUTO) |
| CrExposure_Movie_F | Movie Recording(F Mode) |
| CrExposure_Movie_SQMotion_P | Movie Recording(Slow&Quick Motion(P)) |
| CrExposure_Movie_SQMotion_A | Movie Recording(Slow&Quick Motion(A)) |
| CrExposure_Movie_SQMotion_S | Movie Recording(Slow&Quick Motion(S)) |
| CrExposure_Movie_SQMotion_M | Movie Recording(Slow&Quick Motion(M)) |
| CrExposure_Movie_SQMotion_AUTO | Movie Recording(Slow&Quick Motion(AUTO)) |
| CrExposure_Movie_SQMotion_F | Movie Recording(Slow&Quick Motion(F Mode)) |
| CrExposure_Flash_Off | Flash Off |
| CrExposure_PictureEffect | PictureEffect |
| CrExposure_HiFrameRate_P | High Frame Rate(P) |
| CrExposure_HiFrameRate_A | High Frame Rate(A) |
| CrExposure_HiFrameRate_S | High Frame Rate(S) |
| CrExposure_HiFrameRate_M | High Frame Rate(M) |
| CrExposure_SQMotion_P | S&Q Motion(P) |
| CrExposure_SQMotion_A | S&Q Motion(A) |
| CrExposure_SQMotion_S | S&Q Motion(S) |
| CrExposure_SQMotion_M | S&Q Motion(M) |
| CrExposure_MOVIE | MOVIE |
| CrExposure_STILL | STILL |

| | |
|---|---|
| CrExposure_Movie_F_Mode | Movie F Mode<br><br>Only valid for models that do not support F mode.<br>Do not use. Will be removed in the next release.<br>This value is GetOnly. Cannot be set. |
| CrExposure_F_MovieOrSQMotion | F(Movie or S&Q)<br><br>This value is GetOnly. Cannot be set. |
| CrExposure_Movie_IntervalRec_F | Interval REC(Movie)(F Mode) *1 |
| CrExposure_Movie_IntervalRec_P | Interval REC(Movie)(P) *1 |
| CrExposure_Movie_IntervalRec_A | Interval REC(Movie)(A) *1 |
| CrExposure_Movie_IntervalRec_S | Interval REC(Movie)(S) *1 |
| CrExposure_Movie_IntervalRec_M | Interval REC(Movie)(M) *1 |
| CrExposure_Movie_IntervalRec_AUTO | Interval REC(Movie)(AUTO) *1 |

*1 : Function of the camera is Time Lapse Movie

## CrDeviceProperty_CompressionFileFormatStill

Get/Set the Compression File Format(Still)

Depends on this setting, available settings vary at CrDeviceProperty_FileType.

| Parameter Code | Explanation |
|---|---|
| CrCompressionFileFormat_JPEG | JPEG |
| CrCompressionFileFormat_HEIF_422 | HEIF (4:2:2) |
| CrCompressionFileFormat_HEIF_420 | HEIF (4:2:0) |

## CrDeviceProperty_FileType

Get/Set the File Format(Still)

Before setting this, check if CrDeviceProperty_CompressionFileFormatStill is set properly.

| Parameter Code | Explanation |
|---|---|
| CrFileType_RawJpeg | RAW+JPEG |
| CrFileType_Jpeg | JPEG |
| CrFileType_Raw | RAW |
| CrFileType_RawHeif | RAW+HEIF |
| CrFileType_Heif | HEIF |

## CrDeviceProperty_JpegQuality

Get/Set the JPEG Quality

| Parameter Code | Explanation |
|---|---|
| CrJpegQuality_Light | Light |
| CrJpegQuality_Standard | Standard |
| CrJpegQuality_Fine | Fine |
| CrJpegQuality_ExFine | Extra fine |

## CrDeviceProperty_WhiteBalance

Get/Set the WhiteBalance

| Parameter Code | Explanation |
|---|---|
| CrWhiteBalance_AWB | AWB |
| CrWhiteBalance_Underwater_Auto | Underwater Auto |
| CrWhiteBalance_Daylight | Daylight |
| CrWhiteBalance_Shadow | Shade |
| CrWhiteBalance_Cloudy | Cloudy |
| CrWhiteBalance_Tungsten | Tungsten (Incandescent) |
| CrWhiteBalance_Fluorescent | Fluorescent |
| CrWhiteBalance_Fluorescent_WarmWhite | Fluor::Warm White(-1) |
| CrWhiteBalance_Fluorescent_CoolWhite | Fluor::Cool White(0) |
| CrWhiteBalance_Fluorescent_DayWhite | Fluor::Day White(+1) |
| CrWhiteBalance_Fluorescent_Daylight | Fluor::Daylight White(+2) |
| CrWhiteBalance_Flush | Flush |
| CrWhiteBalance_ColorTemp | C.Temp. |
| CrWhiteBalance_Custom_1 | Custom1 |
| CrWhiteBalance_Custom_2 | Custom2 |
| CrWhiteBalance_Custom_3 | Custom3 |
| CrWhiteBalance_Custom | Custom |

## CrDeviceProperty_FocusMode

Get/Set the Focus Mode

| Parameter Code | Explanation |
|---|---|
| CrFocus_MF | Manual Focus |
| CrFocus_AF_S | Single-shot AF |
| CrFocus_AF_C | Continuous AF |
| CrFocus_AF_A | Automatic AF |
| CrFocus_AF_D | Reserved |
| CrFocus_DMF | Direct Manual Focus |
| CrFocus_PF | Preset Focus |

## CrDeviceProperty_MeteringMode

Get/Set the Exposure Metering Mode

| Parameter Code | Explanation |
|---|---|
| CrMetering_Average | Average |
| CrMetering_CenterWeightedAverage | Center-weighted-average |
| CrMetering_MultiSpot | Multi-spot |
| CrMetering_CenterSpot | Center-spot |
| CrMetering_Multi | Multi |
| CrMetering_CenterWeighted | Center-weighted |
| CrMetering_EntireScreenAverage | Entire Screen Avg. |
| CrMetering_Spot_Standard | Spot : Standard |
| CrMetering_Spot_Large | Spot : Large |
| CrMetering_HighLightWeighted | Highlight |

## CrDeviceProperty_FlashMode

Get/Set the Flash Mode

| Parameter Code | Explanation |
|---|---|
| CrFlash_Auto | Auto flash |
| CrFlash_Off | Flash off |
| CrFlash_Fill | Fill flash |
| CrFlash_ExternalSync | External Sync |
| CrFlash_SlowSync | Slow Sync |
| CrFlash_RearSync | Rear Sync |

## CrDeviceProperty_WirelessFlash

Get/Set the Wireless Flash Setting

| Parameter Code | Explanation |
|---|---|
| CrWirelessFlash_Off | Off |
| CrWirelessFlash_On | On |

## CrDeviceProperty_RedEyeReduction

Get/Set the Red Eye Reduction

| Parameter Code | Explanation |
|---|---|
| CrRedEye_Off | Off |
| CrRedEye_On | On |

## CrDeviceProperty_DriveMode

Get/Set the Drive Mode (Still Capture Mode)

| Parameter Code | Explanation |
|---|---|
| CrDrive_Single | Normal |
| CrDrive_Continuous_Hi | Continuous Shot hi |
| CrDrive_Continuous_Hi_Plus | Cont. Shooting Hi+ |
| CrDrive_Continuous_Hi_Live | Cont. Shooting Hi-Live |
| CrDrive_Continuous_Lo | Continuous Shot lo |
| CrDrive_Continuous | Continuous Shot |
| CrDrive_Continuous_SpeedPriority | Continuous Shot Speed Priority |
| CrDrive_Continuous_Mid | Continuous Shot mid |
| CrDrive_Continuous_Mid_Live | Cont. Shooting Mid-Live |
| CrDrive_Continuous_Lo_Live | Cont. Shooting Lo-Live |
| CrDrive_Timelapse | Timelapse |
| CrDrive_Timer_5s | Self Timer 5sec |
| CrDrive_Timer_10s | Self Timer 10sec |
| CrDrive_Timer_2s | Self Timer 2sec |
| CrDrive_Continuous_Bracket_03Ev_3pics | Continuous Bracket 0.3EV 3pics |
| CrDrive_Continuous_Bracket_03Ev_5pics | Continuous Bracket 0.3EV 5pics |
| CrDrive_Continuous_Bracket_03Ev_9pics | Continuous Bracket 0.3EV 9pics |
| CrDrive_Continuous_Bracket_05Ev_3pics | Continuous Bracket 0.5EV 3pics |
| CrDrive_Continuous_Bracket_05Ev_5pics | Continuous Bracket 0.5EV 5pics |
| CrDrive_Continuous_Bracket_05Ev_9pics | Continuous Bracket 0.5EV 9pics |
| CrDrive_Continuous_Bracket_07Ev_3pics | Continuous Bracket 0.7EV 3pics |
| CrDrive_Continuous_Bracket_07Ev_5pics | Continuous Bracket 0.7EV 5pics |

| | |
|---|---|
| CrDrive_Continuous_Bracket_07Ev_9pics | Continuous Bracket 0.7EV 9pics |
| CrDrive_Continuous_Bracket_10Ev_3pics | Continuous Bracket 1.0EV 3pics |
| CrDrive_Continuous_Bracket_10Ev_5pics | Continuous Bracket 1.0EV 5pics |
| CrDrive_Continuous_Bracket_10Ev_9pics | Continuous Bracket 1.0EV 9pics |
| CrDrive_Continuous_Bracket_20Ev_3pics | Continuous Bracket 2.0EV 3pics |
| CrDrive_Continuous_Bracket_20Ev_5pics | Continuous Bracket 2.0EV 5pics |
| CrDrive_Continuous_Bracket_30Ev_3pics | Continuous Bracket 3.0EV 3pics |
| CrDrive_Continuous_Bracket_30Ev_5pics | Continuous Bracket 3.0EV 5pics |
| CrDrive_Single_Bracket_03Ev_3pics | Single Bracket 0.3EV 3pics |
| CrDrive_Single_Bracket_03Ev_5pics | Single Bracket 0.3EV 5pics |
| CrDrive_Single_Bracket_03Ev_9pics | Single Bracket 0.3EV 9pics |
| CrDrive_Single_Bracket_05Ev_3pics | Single Bracket 0.5EV 3pics |
| CrDrive_Single_Bracket_05Ev_5pics | Single Bracket 0.5EV 5pics |
| CrDrive_Single_Bracket_05Ev_9pics | Single Bracket 0.5EV 9pics |
| CrDrive_Single_Bracket_07Ev_3pics | Single Bracket 0.7EV 3pics |
| CrDrive_Single_Bracket_07Ev_5pics | Single Bracket 0.7EV 5pics |
| CrDrive_Single_Bracket_07Ev_9pics | Single Bracket 0.7EV 9pics |
| CrDrive_Single_Bracket_10Ev_3pics | Single Bracket 1.0EV 3pics |
| CrDrive_Single_Bracket_10Ev_5pics | Single Bracket 1.0EV 5pics |
| CrDrive_Single_Bracket_10Ev_9pics | Single Bracket 1.0EV 9pics |
| CrDrive_Single_Bracket_20Ev_3pics | Single Bracket 2.0EV 3pics |
| CrDrive_Single_Bracket_20Ev_5pics | Single Bracket 2.0EV 5pics |
| CrDrive_Single_Bracket_30Ev_3pics | Single Bracket 3.0EV 3pics |
| CrDrive_Single_Bracket_30Ev_5pics | Single Bracket 3.0EV 5pics |
| CrDrive_WB_Bracket_Lo | WhiteBalance Bracket Lo |
| CrDrive_WB_Bracket_Hi | WhiteBalance Bracket Hi |
| CrDrive_DRO_Bracket_Lo | DRO Bracket Lo |
| CrDrive_DRO_Bracket_Hi | DRO Bracket Hi |
| CrDrive_LPF_Bracket | LPF Bracket |
| CrDrive_RemoteCommander | Remote Commander |
| CrDrive_MirrorUp | Mirror Up |
| CrDrive_SelfPortrait_1 | Self Portrait 1 Person |
| CrDrive_SelfPortrait_2 | Self Portrait 2people |
| CrDrive_Continuous_Timer_3pics | Continuous Self Timer 3pics |
| CrDrive_Continuous_Timer_5pics | Continuous Self Timer 5pics |
| CrDrive_Continuous_Timer_5s_3pics | Continuous Self Timer 3pics 5sec |
| CrDrive_Continuous_Timer_5s_5pics | Continuous Self Timer 5pics 5sec |
| CrDrive_Continuous_Timer_2s_3pics | Continuous Self Timer 3pics 2sec |
| CrDrive_Continuous_Timer_2s_5pics | Continuous Self Timer 5pics 2sec |
| CrDrive_SingleBurstShooting_lo | Spot Burst Shooting Lo |
| CrDrive_SingleBurstShooting_mid | Spot Burst Shooting Mid |
| CrDrive_SingleBurstShooting_hi | Spot Burst Shooting Hi |

| | |
|---|---|
| CrDrive_Continuous_Bracket_03Ev_2pics_Plus | Continuous Bracket 0.3EV 2pics+ |
| CrDrive_Continuous_Bracket_03Ev_2pics_Minus | Continuous Bracket 0.3EV 2pics- |
| CrDrive_Continuous_Bracket_03Ev_7pics | Continuous Bracket 0.3EV 7pics |
| CrDrive_Continuous_Bracket_05Ev_2pics_Plus | Continuous Bracket 0.5EV 2pics+ |
| CrDrive_Continuous_Bracket_05Ev_2pics_Minus | Continuous Bracket 0.5EV 2pics- |
| CrDrive_Continuous_Bracket_05Ev_7pics | Continuous Bracket 0.5EV 7pics |
| CrDrive_Continuous_Bracket_07Ev_2pics_Plus | Continuous Bracket 0.7EV 2pics+ |
| CrDrive_Continuous_Bracket_07Ev_2pics_Minus | Continuous Bracket 0.7EV 2pics- |
| CrDrive_Continuous_Bracket_07Ev_7pics | Continuous Bracket 0.7EV 7pics |
| CrDrive_Continuous_Bracket_10Ev_2pics_Plus | Continuous Bracket 1.0EV 2pics+ |
| CrDrive_Continuous_Bracket_10Ev_2pics_Minus | Continuous Bracket 1.0EV 2pics- |
| CrDrive_Continuous_Bracket_10Ev_7pics | Continuous Bracket 1.0EV 7pics |
| CrDrive_Continuous_Bracket_13Ev_2pics_Plus | Continuous Bracket 1.3EV 2pics+ |
| CrDrive_Continuous_Bracket_13Ev_2pics_Minus | Continuous Bracket 1.3EV 2pics- |
| CrDrive_Continuous_Bracket_13Ev_3pics | Continuous Bracket 1.3EV 3pics |
| CrDrive_Continuous_Bracket_13Ev_5pics | Continuous Bracket 1.3EV 5pics |
| CrDrive_Continuous_Bracket_13Ev_7pics | Continuous Bracket 1.3EV 7pics |
| CrDrive_Continuous_Bracket_15Ev_2pics_Plus | Continuous Bracket 1.5EV 2pics+ |
| CrDrive_Continuous_Bracket_15Ev_2pics_Minus | Continuous Bracket 1.5EV 2pics- |
| CrDrive_Continuous_Bracket_15Ev_3pics | Continuous Bracket 1.5EV 3pics |
| CrDrive_Continuous_Bracket_15Ev_5pics | Continuous Bracket 1.5EV 5pics |
| CrDrive_Continuous_Bracket_15Ev_7pics | Continuous Bracket 1.7EV 7pics |
| CrDrive_Continuous_Bracket_17Ev_2pics_Plus | Continuous Bracket 1.7EV 2pics+ |
| CrDrive_Continuous_Bracket_17Ev_2pics_Minus | Continuous Bracket 1.7EV 2pics- |
| CrDrive_Continuous_Bracket_17Ev_3pics | Continuous Bracket 1.7EV 3pics |
| CrDrive_Continuous_Bracket_17Ev_5pics | Continuous Bracket 1.7EV 5pics |
| CrDrive_Continuous_Bracket_17Ev_7pics | Continuous Bracket 1.7EV 7pics |
| CrDrive_Continuous_Bracket_20Ev_2pics_Plus | Continuous Bracket 2.0EV 2pics+ |
| CrDrive_Continuous_Bracket_20Ev_2pics_Minus | Continuous Bracket 2.0EV 2pics- |
| CrDrive_Continuous_Bracket_20Ev_7pics | Continuous Bracket 2.0EV 7pics |
| CrDrive_Continuous_Bracket_23Ev_2pics_Plus | Continuous Bracket 2.3EV 2pics+ |
| CrDrive_Continuous_Bracket_23Ev_2pics_Minus | Continuous Bracket 2.3EV 2pics- |
| CrDrive_Continuous_Bracket_23Ev_3pics | Continuous Bracket 2.3EV 3pics |
| CrDrive_Continuous_Bracket_23Ev_5pics | Continuous Bracket 2.3EV 5pics |
| CrDrive_Continuous_Bracket_25Ev_2pics_Plus | Continuous Bracket 2.5EV 2pics+ |
| CrDrive_Continuous_Bracket_25Ev_2pics_Minus | Continuous Bracket 2.5EV 2pics- |
| CrDrive_Continuous_Bracket_25Ev_3pics | Continuous Bracket 2.5EV 3pics |
| CrDrive_Continuous_Bracket_25Ev_5pics | Continuous Bracket 2.5EV 5pics |
| CrDrive_Continuous_Bracket_27Ev_2pics_Plus | Continuous Bracket 2.7EV 2pics+ |
| CrDrive_Continuous_Bracket_27Ev_2pics_Minus | Continuous Bracket 2.7EV 2pics- |
| CrDrive_Continuous_Bracket_27Ev_3pics | Continuous Bracket 2.7EV 3pics |
| CrDrive_Continuous_Bracket_27Ev_5pics | Continuous Bracket 2.7EV 5pics |

| | |
|---|---|
| CrDrive_Continuous_Bracket_30Ev_2pics_Plus | Continuous Bracket 3.0EV 2pics+ |
| CrDrive_Continuous_Bracket_30Ev_2pics_Minus | Continuous Bracket 3.0EV 2pics- |
| CrDrive_Single_Bracket_03Ev_2pics_Plus | Single Bracket 0.3EV 2pics+ |
| CrDrive_Single_Bracket_03Ev_2pics_Minus | Single Bracket 0.3EV 2pics- |
| CrDrive_Single_Bracket_03Ev_7pics | Single Bracket 0.3EV 7pics |
| CrDrive_Single_Bracket_05Ev_2pics_Plus | Single Bracket 0.5EV 2pics+ |
| CrDrive_Single_Bracket_05Ev_2pics_Minus | Single Bracket 0.5EV 2pics- |
| CrDrive_Single_Bracket_05Ev_7pics | Single Bracket 0.5EV 7pics |
| CrDrive_Single_Bracket_07Ev_2pics_Plus | Single Bracket 0.7EV 2pics+ |
| CrDrive_Single_Bracket_07Ev_2pics_Minus | Single Bracket 0.7EV 2pics- |
| CrDrive_Single_Bracket_07Ev_7pics | Single Bracket 0.7EV 7pics |
| CrDrive_Single_Bracket_10Ev_2pics_Plus | Single Bracket 1.0EV 2pics+ |
| CrDrive_Single_Bracket_10Ev_2pics_Minus | Single Bracket 1.0EV 2pics- |
| CrDrive_Single_Bracket_10Ev_7pics | Single Bracket 1.0EV 7pics |
| CrDrive_Single_Bracket_13Ev_2pics_Plus | Single Bracket 1.3EV 2pics+ |
| CrDrive_Single_Bracket_13Ev_2pics_Minus | Single Bracket 1.3EV 2pics- |
| CrDrive_Single_Bracket_13Ev_3pics | Single Bracket 1.3EV 3pics |
| CrDrive_Single_Bracket_13Ev_5pics | Single Bracket 1.3EV 5pics |
| CrDrive_Single_Bracket_13Ev_7pics | Single Bracket 1.3EV 7pics |
| CrDrive_Single_Bracket_15Ev_2pics_Plus | Single Bracket 1.5EV 2pics+ |
| CrDrive_Single_Bracket_15Ev_2pics_Minus | Single Bracket 1.5EV 2pics- |
| CrDrive_Single_Bracket_15Ev_3pics | Single Bracket 1.5EV 3pics |
| CrDrive_Single_Bracket_15Ev_5pics | Single Bracket 1.5EV 5pics |
| CrDrive_Single_Bracket_15Ev_7pics | Single Bracket 1.5EV 7pics |
| CrDrive_Single_Bracket_17Ev_2pics_Plus | Single Bracket 1.7EV 2pics+ |
| CrDrive_Single_Bracket_17Ev_2pics_Minus | Single Bracket 1.7EV 2pics- |
| CrDrive_Single_Bracket_17Ev_3pics | Single Bracket 1.7EV 3pics |
| CrDrive_Single_Bracket_17Ev_5pics | Single Bracket 1.7EV 5pics |
| CrDrive_Single_Bracket_17Ev_7pics | Single Bracket 1.7EV 7pics |
| CrDrive_Single_Bracket_20Ev_2pics_Plus | Single Bracket 2.0EV 2pics+ |
| CrDrive_Single_Bracket_20Ev_2pics_Minus | Single Bracket 2.0EV 2pics- |
| CrDrive_Single_Bracket_20Ev_7pics | Single Bracket 2.0EV 7pics |
| CrDrive_Single_Bracket_23Ev_2pics_Plus | Single Bracket 2.3EV 2pics+ |
| CrDrive_Single_Bracket_23Ev_2pics_Minus | Single Bracket 2.3EV 2pics- |
| CrDrive_Single_Bracket_23Ev_3pics | Single Bracket 2.3EV 3pics |
| CrDrive_Single_Bracket_23Ev_5pics | Single Bracket 2.3EV 5pics |
| CrDrive_Single_Bracket_25Ev_2pics_Plus | Single Bracket 2.5EV 2pics+ |
| CrDrive_Single_Bracket_25Ev_2pics_Minus | Single Bracket 2.5EV 2pics- |
| CrDrive_Single_Bracket_25Ev_3pics | Single Bracket 2.5EV 3pics |
| CrDrive_Single_Bracket_25Ev_5pics | Single Bracket 2.5EV 5pics |
| CrDrive_Single_Bracket_27Ev_2pics_Plus | Single Bracket 2.7EV 2pics+ |
| CrDrive_Single_Bracket_27Ev_2pics_Minus | Single Bracket 2.7EV 2pics- |

| | |
|---|---|
| CrDrive_Single_Bracket_27Ev_3pics | Single Bracket 2.7EV 3pics |
| CrDrive_Single_Bracket_27Ev_5pics | Single Bracket 2.7EV 5pics |
| CrDrive_Single_Bracket_30Ev_2pics_Plus | Single Bracket 3.0EV 2pics+ |
| CrDrive_Single_Bracket_30Ev_2pics_Minus | Single Bracket 3.0EV 2pics- |
| CrDrive_FocusBracket | Focus Bracket |

## CrDeviceProperty_DRO

Get/Set the Dynamic Range Optimizer

| Parameter Code | Explanation |
|---|---|
| CrDRangeOptimizer_Off | DRO OFF |
| CrDRangeOptimizer_On | DRO |
| CrDRangeOptimizer_Plus | DRO+ |
| CrDRangeOptimizer_Plus_Manual_1 | DRO + Manual1 |
| CrDRangeOptimizer_Plus_Manual_2 | DRO + Manual2 |
| CrDRangeOptimizer_Plus_Manual_3 | DRO + Manual3 |
| CrDRangeOptimizer_Plus_Manual_4 | DRO + Manual4 |
| CrDRangeOptimizer_Plus_Manual_5 | DRO + Manual5 |
| CrDRangeOptimizer_Auto | DRO AUTO |
| CrDRangeOptimizer_HDR_Auto | HDR AUTO |
| CrDRangeOptimizer_HDR_10Ev | HDR 1.0Ev |
| CrDRangeOptimizer_HDR_20Ev | HDR 2.0Ev |
| CrDRangeOptimizer_HDR_30Ev | HDR 3.0Ev |
| CrDRangeOptimizer_HDR_40Ev | HDR 4.0Ev |
| CrDRangeOptimizer_HDR_50Ev | HDR 5.0Ev |
| CrDRangeOptimizer_HDR_60Ev | HDR 6.0Ev |

## CrDeviceProperty_ImageSize

Get/Set the Image Size

| Parameter Code | Explanation |
|---|---|
| CrImageSize_L | L |
| CrImageSize_M | M |
| CrImageSize_S | S |
| CrImageSize_VGA | VGA |

## CrDeviceProperty_AspectRatio

Get/Set the Aspect Ratio

| Parameter Code | Explanation |
|---|---|
| CrAspectRatio_3_2 | 3:2 |
| CrAspectRatio_16_9 | 16:9 |
| CrAspectRatio_4_3 | 4:3 |
| CrAspectRatio_1_1 | 1:1 |

## CrDeviceProperty_PictureEffect

Get/Set the Picture Effect Value

| Parameter Code | Explanation |
|---|---|
| CrPictureEffect_Off | OFF |
| CrPictureEffect_ToyCameraNormal | Toy Camera Normal |
| CrPictureEffect_ToyCameraCool | Toy Camera Cool |
| CrPictureEffect_ToyCameraWarm | Toy Camera Warm |
| CrPictureEffect_ToyCameraGreen | Toy Camera Green |
| CrPictureEffect_ToyCameraMagenta | Toy Camera Magenta |
| CrPictureEffect_Pop | Pop Color |
| CrPictureEffect_PosterizationBW | Posterization B/W |
| CrPictureEffect_PosterizationColor | Posterization Color |
| CrPictureEffect_Retro | Retro Photo |
| CrPictureEffect_SoftHighkey | Soft High-key |
| CrPictureEffect_PartColorRed | Partial Color Red |
| CrPictureEffect_PartColorGreen | Partial Color Green |
| CrPictureEffect_PartColorBlue | Partial Color Blue |
| CrPictureEffect_PartColorYellow | Partial Color Yellow |
| CrPictureEffect_HighContrastMonochrome | High Contrast Mono |
| CrPictureEffect_SoftFocusLow | Soft Focus Low |
| CrPictureEffect_SoftFocusMid | Soft Focus Mid |

| | |
|---|---|
| CrPictureEffect_SoftFocusHigh | Soft Focus High |
| CrPictureEffect_HDRPaintingLow | HDR Painting Low |
| CrPictureEffect_HDRPaintingMid | HDR Painting Mid |
| CrPictureEffect_HDRPaintingHigh | HDR Painting High |
| CrPictureEffect_RichToneMonochrome | Rich-tone Mono |
| CrPictureEffect_MiniatureAuto | Miniature Auto |
| CrPictureEffect_MiniatureTop | Miniature Top |
| CrPictureEffect_MiniatureMidHorizontal | Miniature Middle(Horizontal) |
| CrPictureEffect_MiniatureBottom | Miniature Bottom |
| CrPictureEffect_MiniatureLeft | Miniature Left |
| CrPictureEffect_MiniatureMidVertical | Miniature Middle(Vertical) |
| CrPictureEffect_MiniatureRight | Miniature Right |
| CrPictureEffect_MiniatureWaterColor | Miniature Water Color |
| CrPictureEffect_MiniatureIllustrationLow | Miniature Illustration Low |
| CrPictureEffect_MiniatureIllustrationMid | Miniature Illustration Mid |
| CrPictureEffect_MiniatureIllustrationHigh | Miniature Illustration High |

## CrDeviceProperty_Colortemp

Get/Set the Color Temperature

For models that support CrDeviceProperty_ColortempStep, the CurrentValue of this device property is also updated by manipulating CrDeviceProperty_ColortempStep.

| Value | Explanation | |
|---|---|---|
| Variable | min | The resolution of the CurrentValue is the step value. The CurrentValue increases or decreases with each step value.<br><br>Ex.)<br>If min = 1000, max = 1500, step = 100, you can set 6 values of 1000, 1100, 1200, 1300, 1400, 1500 to CurrentValue.<br><br>The special CurrentValue are following.<br>- 0x0000 means less than min.<br>- 0xFFFF means greater than max.<br>These value is not included the value of Range. (It is only used as CurrentValue.)<br><br>Note:<br>In ILME-FX6, it is always GetOnly, regardless of the return value of IsSetEnableCurrentValue(). |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ColorTuningAB

Get/Set the Biaxial Fine Tuning A-B Direction

| Value | Explanation | |
|---|---|---|
| 0x9C(B9_00) | min | AB value sent to PC App from camera corresponds to one of the following patterns.<br>AB number is BY or AY, where Y is decimal from 0.00 to 9.00 and increments by 0.25.<br><br>Ex.)<br>B9.00(0x9C), B8.75(0x9D), ..., A8.75(0xE3), A9.00(0xE4).<br><br>Note:<br>There may be parameter scope differences due to model differences. |
| 0xE4(A9_00) | max | |
| 0x01(0.25) | step | |

## CrDeviceProperty_ColorTuningGM

Get/Set the Biaxial Fine Tuning G-M Direction

| Value | Explanation | |
|---|---|---|
| 0x9C(M9_00) | min | GM value sent to PC App from camera corresponds to one of the following patterns.<br>GM number is MX or GX, where X is decimal from 0.00 to 9.00 and increments by 0.25.<br><br>Ex.)<br>M9.00(0x9C), M8.75(0x9D), ..., G8.75(0xE3), G9.00(0xE4).<br><br>Note:<br>There may be parameter differences due to model differences. |
| 0xE4(G9_00) | max | |
| 0x01(0.25) | step | |

## CrDeviceProperty_LiveViewDisplayEffect

Get/Set the Live View Display Effect

| Parameter Code | Explanation |
|---|---|
| CrLiveViewDisplayEffect_Unknown | Unknown |
| CrLiveViewDisplayEffect_ON | Effect ON |
| CrLiveViewDisplayEffect_OFF | Effect OFF |

## CrDeviceProperty_StillImageStoreDestination

Get the information of Still Image Save Destination

| Parameter Code | Explanation |
|---|---|
| CrStillImageStoreDestination_HostPC | Host Device (Ex. PC) |
| CrStillImageStoreDestination_MemoryCard | Camera(Memory Card) |
| CrStillImageStoreDestination_HostPCAndMemoryCard | Host Device & Camera(Memory Card) |

## CrDeviceProperty_PriorityKeySettings

Get/Set the Position Key Setting

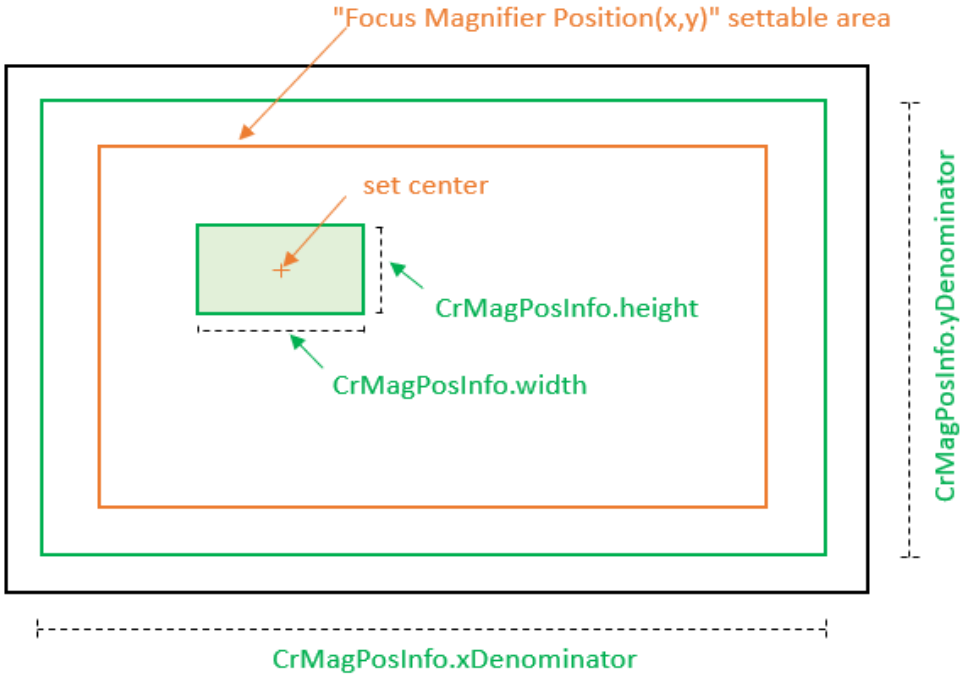
| Parameter Code | Explanation |
|---|---|
| CrPriorityKey_CameraPosition | Camera position priority<br>(Ex. Mode dial, Drive/Focus mode dial) |
| CrPriorityKey_PCRemote | PC Remote setting priority |

## CrDeviceProperty_Focus_Magnifier_Setting

Get/Set the Focus Magnifier Setting

| Value | Explanation |
|---|---|
| 0x0000000000000000<br><br>~<br><br>0xFFFFFFFFFFFFFFFF | The upper 4 bytes are the Focus Magnifier Ratio, and the lower 4 bytes are the Focus Magnifier Position(x,y).<br><br>Caution :<br>The range of focus magnifier ratio and focus magnifier position varies depending on the model and aspect ratio.<br><br>[Upper 4bytes] Regarding Focus Magnifier Ratio :<br>Select the focus magnifier ratio to be set from the focus magnifier ratio obtained by GetValues() function.<br><br>Ex.) Result obtained by GetValues() function.<br>If the camera supports OFF, x1.0, x4.0 and x8.0 as focus magnifier ratio, Result is the following.<br>Enum value[0] = 0x00000000FFFFFFFF (means OFF)<br>Enum value[1] = 0x0000000AFFFFFFFF (means x1.0)<br>Enum value[2] = 0x00000028FFFFFFFF (means x4.0)<br>Enum value[3] = 0x00000050FFFFFFFF (means x8.0)<br><br>[Lower 4bytes] Regarding Focus Magnifier Position (x,y) :<br>The upper 2 bytes are the x coordinate and the lower 2 bytes are the y coordinate.<br>If this part is 0xFFFFFFFF, it means an invalid value.<br>If focus magnifier position (x) is 150 and (y) is 100, set 0x00960064. 0x0096 = 0d150, 0x0064 = 0d100.<br>The range of X is 0 ~ 639 (0x027F), and the range of Y is 0 ~ 479 (0x01DF).<br>Frame size is acquired by CrMagPosInfo. CrMagPosInfo is in LiveViewProperty.<br>Since this position specifies the center of the frame, the position range is more inside by half the frame size than CrMagPosInfo.xDenominator/yDenominator.<br><br>Caution:<br>If it is not in the magnified focus state, the desired result may not be obtained unless the correct position is set again after refreshing the state by setting 0xFFFFFFFF (Invalid Value) in the lower 4 bytes in advance.<br><br>Note:<br>See Tips/Trouble shooting for a detailed implementation example.<br>Focus Magnifier Setting |

Fig. Relationship between CrMagPosInfo and settable area



## CrDeviceProperty_DateTime_Settings

Set the Date and Time

| Parameter Code | Explanation |
|---|---|
| - | 64bit value.<br><br>Specify the time in UNIX time (elapsed time from 1970/01/01 00:00:00).The time displayed is linked to the time zone setting of the camera. The range depends on the model and firmware.<br><br>Ex.) when 1609582830 is set<br>= 2021/01/02 10:20:30(UTC)<br>= 2021/01/02 19:20:30(Tokyo) |

## CrDeviceProperty_NearFar

Get the Focus Near/Far Enable Status

| Parameter Code | Explanation |
|---|---|
| CrNearFar_Disable | Disable |
| CrNearFar_Enable | Enable |

Set the Focus Near/Far

| Value | Explanation |
|---|---|
| -7 | min<br><br>Specify to change the focus to Near.<br><br>Can be set from -1 to -7 in steps. Larger value makes the movement width larger. *1 |
| 7 | max<br><br>Specify to change the focus to Far.<br><br>Can be set in steps of 1 to 7. Larger value makes the movement width larger. *1 |
| 1 | step |

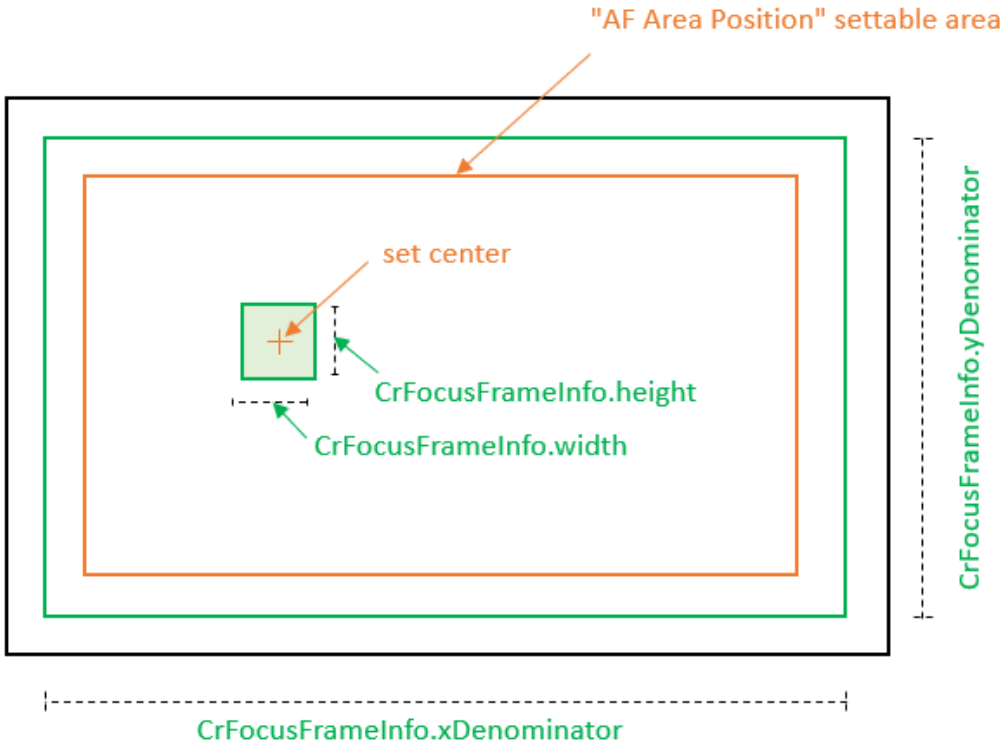
*1 : In the case of DSC-RX0M2, the movement width is fixed.

## CrDeviceProperty_AF_Area_Position

Set the AF Area Position(x,y)

| Value | Explanation |
|---|---|
| 0x00000000 ~ 0xFFFFFFFF | Set the center position of the AF frame.<br><br>The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes<br>The range of X is 0 ~ 639 (0x027F), and the range of Y is 0 ~ 479 (0x01DF).<br><br>AF frame size is acquired by CrFocusFrameInfo.<br>CrFocusFrameInfo is in LiveViewProperty.<br><br>The settable area is more inside by half the frame size than CrFocusFrameInfo.xDenominator/yDenominator.<br><br>Note:<br>The range in which the coordinates can be specified varies depending on the model, aspect setting, and AF setting. |

Fig.Relationship between CrFocusframeInfo and settable area



## CrDeviceProperty_Zoom_Scale

Get/Set the Zoom Scale.
It may not be possible to operate depending on the model and lens type. refs Zoom Operation / Zoom Scale.

| Value | Explanation | |
|---|---|---|
| Variable | min | min/max/Value should be set in units of “step”.<br><br>Ex.) min: 1000, max: 8000, step: 200, value: 1200 (min = x1.0, max = x8.0, value = x1.2) |
| Variable | max | |
| Variable | step<br><br>This value varies depending on the camera’s configurable conditions. (in units of 0.001) | |

## CrDeviceProperty_Zoom_Setting

Get/Set the Zoom Setting.
It may not be possible to operate depending on the model and lens type. refs Zoom Operation / Zoom Scale.

| Parameter Code | Explanation |
|---|---|
| CrZoomSetting_OpticalZoomOnly | Optical zoom only |
| CrZoomSetting_SmartZoomOnly | Smart zoom only |
| CrZoomSetting_On_ClearImageZoom | Clear image zoom on |
| CrZoomSetting_On_DigitalZoom | Digital zoom (and Clear image zoom) on |

## CrDeviceProperty_Zoom_Operation

Execute the Zoom Operation.
It may not be possible to operate depending on the model and lens type. refs Zoom Operation / Zoom Scale.
For models that support CrDeviceProperty_Zoom_Speed_Range, link with Range(min/max/step) of CrDeviceProperty_Zoom_Speed_Range.

| Parameter Code | Explanation | |
|---|---|---|
| Variable<br><br>(Negative number) | min | Zoom out (-)<br>Default value is CrZoomOperation_Wide.<br>When you specify zoom out, the zoom out continues until it “Zoom stop” or until the lens or setting limit is reached. |
| 0 (Zero) | - | Zoom stop<br>You can use the CrZoomOperation_Stop. |
| Variable<br><br>(Positive number) | max | Zoom in (+)<br>Default value is CrZoomOperation_Tele.<br>When you specify zoom in, the zoom in continues until it “Zoom stop” or until the lens or setting limit is reached. |

Note: ILME-FX3 and ILME-FX30 does not support CrDeviceProperty_Zoom_Speed_Range, but you can change the zoom speed from -8 to +8.

## CrDeviceProperty_Movie_File_Format

Get/Set the File Format(Movie)

| Parameter Code | Explanation |
|---|---|
| CrFileFormatMovie_AVCHD | AVCHD |
| CrFileFormatMovie_MP4 | MP4 |
| CrFileFormatMovie_XAVC_S_4K | XAVC S 4K |
| CrFileFormatMovie_XAVC_S_HD | XAVC S HD |
| CrFileFormatMovie_XAVC_HS_8K | XAVC HS 8K |
| CrFileFormatMovie_XAVC_HS_4K | XAVC HS 4K |
| CrFileFormatMovie_XAVC_S_L_4K | XAVC S-L 4K |
| CrFileFormatMovie_XAVC_S_L_HD | XAVC S-L HD |
| CrFileFormatMovie_XAVC_S_I_4K | XAVC S-I 4K |
| CrFileFormatMovie_XAVC_S_I_HD | XAVC S-I HD |
| CrFileFormatMovie_XAVC_I | XAVC I |
| CrFileFormatMovie_XAVC_L | XAVC L |
| CrFileFormatMovie_XAVC_HS_HD | XAVC HS HD |
| CrFileFormatMovie_XAVC_S_I_DCI_4K | XAVC S-I DCI 4K |

Note: In some models, “XAVC S-L xx” is displayed as “XAVC S xx” in their menu.

## CrDeviceProperty_Movie_Recording_Setting

Get/Set the Recording Setting(Movie)

| Parameter Code | Explanation |
|---|---|
| CrRecordingSettingMovie_60p_50M | 60p 50M / XAVC S |
| CrRecordingSettingMovie_30p_50M | 30p 50M / XAVC S |
| CrRecordingSettingMovie_24p_50M | 24p 50M / XAVC S |
| CrRecordingSettingMovie_50p_50M | 50p 50M / XAVC S |
| CrRecordingSettingMovie_25p_50M | 25p 50M / XAVC S |
| CrRecordingSettingMovie_60i_24M | 60i 24M(FX) / AVCHD |
| CrRecordingSettingMovie_50i_24M_FX | 50i 24M(FX) / AVCHD |
| CrRecordingSettingMovie_60i_17M_FH | 60i 17M(FH) / AVCHD |
| CrRecordingSettingMovie_50i_17M_FH | 50i 17M(FH) / AVCHD |
| CrRecordingSettingMovie_60p_28M_PS | 60p 28M(PS) / AVCHD |
| CrRecordingSettingMovie_50p_28M_PS | 50p 28M(PS) / AVCHD |
| CrRecordingSettingMovie_24p_24M_FX | 24p 24M(FX) / AVCHD |
| CrRecordingSettingMovie_25p_24M_FX | 25p 24M(FX) / AVCHD |
| CrRecordingSettingMovie_24p_17M_FH | 24p 17M(FH) / AVCHD |
| CrRecordingSettingMovie_25p_17M_FH | 25p 17M(FH) / AVCHD |

| | |
|---|---|
| CrRecordingSettingMovie_120p_50M_1280x720 | 120p 50M (1280x720) / XAVC S |
| CrRecordingSettingMovie_100p_50M_1280x720 | 100p 50M (1280x720) / XAVC S |
| CrRecordingSettingMovie_1920x1080_30p_16M | 1920x1080 30p 16M / MP4 |
| CrRecordingSettingMovie_1920x1080_25p_16M | 1920x1080 25p 16M / MP4 |
| CrRecordingSettingMovie_1280x720_30p_6M | 1280x720 30p 6M / MP4 |
| CrRecordingSettingMovie_1280x720_25p_6M | 1280x720 25p 6M / MP4 |
| CrRecordingSettingMovie_1920x1080_60p_28M | 1920x1080 60p 28M / MP4 |
| CrRecordingSettingMovie_1920x1080_50p_28M | 1920x1080 50p 28M / MP4 |
| CrRecordingSettingMovie_60p_25M_XAVC_S_HD | 60p 25M / XAVC S HD |
| CrRecordingSettingMovie_50p_25M_XAVC_S_HD | 50p 25M / XAVC S HD |
| CrRecordingSettingMovie_30p_16M_XAVC_S_HD | 30p 16M / XAVC S HD |
| CrRecordingSettingMovie_25p_16M_XAVC_S_HD | 25p 16M / XAVC S HD |
| CrRecordingSettingMovie_120p_100M_1920x 1080_XAVC_S_HD | 120p 100M (1920x1080) / XAVC S HD |
| CrRecordingSettingMovie_100p_100M_1920x 1080_XAVC_S_HD | 100p 100M (1920x1080) / XAVC S HD |
| CrRecordingSettingMovie_120p_60M_1920x 1080_XAVC_S_HD | 120p 60M (1920x1080) / XAVC S HD |
| CrRecordingSettingMovie_100p_60M_1920x 1080_XAVC_S_HD | 100p 60M (1920x1080) / XAVC S HD |
| CrRecordingSettingMovie_30p_100M_XAVC_S_4K | 30p 100M / XAVC S 4K |
| CrRecordingSettingMovie_25p_100M_XAVC_S_4K | 25p 100M / XAVC S 4K |
| CrRecordingSettingMovie_24p_100M_XAVC_S_4K | 24p 100M / XAVC S 4K |
| CrRecordingSettingMovie_30p_60M_XAVC_S_4K | 30p 60M / XAVC S 4K |
| CrRecordingSettingMovie_25p_60M_XAVC_S_4K | 25p 60M / XAVC S 4K |
| CrRecordingSettingMovie_24p_60M_XAVC_S_4K | 24p 60M / XAVC S 4K |
| CrRecordingSettingMovie_600M_422_10bit | 600M 422 10bit |
| CrRecordingSettingMovie_500M_422_10bit | 500M 422 10bit |
| CrRecordingSettingMovie_400M_420_10bit | 400M 420 10bit |
| CrRecordingSettingMovie_300M_422_10bit | 300M 422 10bit |
| CrRecordingSettingMovie_280M_422_10bit | 280M 422 10bit |
| CrRecordingSettingMovie_250M_422_10bit | 250M 422 10bit |
| CrRecordingSettingMovie_240M_422_10bit | 240M 422 10bit |
| CrRecordingSettingMovie_222M_422_10bit | 222M 422 10bit |
| CrRecordingSettingMovie_200M_422_10bit | 200M 422 10bit |
| CrRecordingSettingMovie_200M_420_10bit | 200M 420 10bit |
| CrRecordingSettingMovie_200M_420_8bit | 200M 420 8bit |
| CrRecordingSettingMovie_185M_422_10bit | 185M 422 10bit |
| CrRecordingSettingMovie_150M_420_10bit | 150M 420 10bit |
| CrRecordingSettingMovie_150M_420_8bit | 150M 420 8bit |
| CrRecordingSettingMovie_140M_422_10bit | 140M 422 10bit |

| | |
|---|---|
| CrRecordingSettingMovie_111M_422_10bit | 111M 422 10bit |
| CrRecordingSettingMovie_100M_422_10bit | 100M 422 10bit |
| CrRecordingSettingMovie_100M_420_10bit | 100M 420 10bit |
| CrRecordingSettingMovie_100M_420_8bit | 100M 420 8bit |
| CrRecordingSettingMovie_93M_422_10bit | 93M 422 10bit |
| CrRecordingSettingMovie_89M_422_10bit | 89M 422 10bit |
| CrRecordingSettingMovie_75M_420_10bit | 75M 420 10bit |
| CrRecordingSettingMovie_60M_420_8bit | 60M 420 8bit |
| CrRecordingSettingMovie_50M_422_10bit | 50M 422 10bit |
| CrRecordingSettingMovie_50M_420_10bit | 50M 420 10bit |
| CrRecordingSettingMovie_50M_420_8bit | 50M 420 8bit |
| CrRecordingSettingMovie_45M_420_10bit | 45M 420 10bit |
| CrRecordingSettingMovie_30M_420_10bit | 30M 420 10bit |
| CrRecordingSettingMovie_25M_420_8bit | 25M 420 8bit |
| CrRecordingSettingMovie_16M_420_8bit | 16M 420 8bit |
| CrRecordingSettingMovie_520M_422_10bit | 520M 422 10bit |
| CrRecordingSettingMovie_260M_422_10bit | 260M 422 10bit |

## CrDeviceProperty_Movie_Recording_FrameRateSetting

Get/Set the Recording Frame Rate Setting(Movie)

| Parameter Code | Explanation |
|---|---|
| CrRecordingFrameRateSettingMovie_120p | 120p<br>Actual frequency might be 119.88. |
| CrRecordingFrameRateSettingMovie_100p | 100p |
| CrRecordingFrameRateSettingMovie_60p | 60p<br>Actual frequency might be 59.94. |
| CrRecordingFrameRateSettingMovie_50p | 50p |
| CrRecordingFrameRateSettingMovie_30p | 30p<br>Actual frequency might be 29.97. |
| CrRecordingFrameRateSettingMovie_25p | 25p |
| CrRecordingFrameRateSettingMovie_24p | 24p<br>Actual frequency might be 23.98 except ILME-FX6. |
| CrRecordingFrameRateSettingMovie_23_98p | 23.98p |
| CrRecordingFrameRateSettingMovie_29_97p | 29.97p |
| CrRecordingFrameRateSettingMovie_59_94p | 59.94p |
| CrRecordingFrameRateSettingMovie_24_00p | 24.00p |
| CrRecordingFrameRateSettingMovie_119_88p | 119.88p |

The value of this Device Property has been changed since version 1.08.00 to be expressed as an exact value.
When using these models, please refer to the table and replace the definitions.

Table fr-1. Target Model list

| No. | Model Name | FW version |
|---|---|---|
| 1 | ILME-FX3 | Ver. 3.00 or later |
| 2 | ILME-FX30 | Ver. 2.00 or later |

Table fr-2. Replacement table

| | version | | | |
|---|---|---|---|---|
| | ~1.07.00 | | 1.08.00 | |
| Enumeration | ILME-FX6 | Other | ILME-FX30<br>ILME-FX3 | Other |
| CrRecordingFrameRateSettingMovie_120p | - | ✓ | - | Same as version 1.07.00 |
| CrRecordingFrameRateSettingMovie_100p | - | ✓ | ✓ | |
| CrRecordingFrameRateSettingMovie_60p | - | ✓ | - | |
| CrRecordingFrameRateSettingMovie_50p | ✓ | ✓ | ✓ | |
| CrRecordingFrameRateSettingMovie_30p | - | ✓ | - | |
| CrRecordingFrameRateSettingMovie_25p | ✓ | ✓ | ✓ | |
| CrRecordingFrameRateSettingMovie_24p | ✓ | ✓ | - | |
| CrRecordingFrameRateSettingMovie_23_98p | ✓ | - | ✓ | |
| CrRecordingFrameRateSettingMovie_29_97p | ✓ | - | ✓ | |
| CrRecordingFrameRateSettingMovie_59_94p | ✓ | - | ✓ | |
| CrRecordingFrameRateSettingMovie_24_00p (NEW) | - | - | ✓ | |
| CrRecordingFrameRateSettingMovie_119_88p (NEW) | - | - | ✓ | |

## CrDeviceProperty_Interval_Rec_Mode

Get the Interval REC Mode

| Parameter Code | Explanation |
|---|---|
| CrIntervalRecMode_OFF | OFF |
| CrIntervalRecMode_ON | ON |

## CrDeviceProperty_Still_Image_Trans_Size

Get/Set the Still Image Trans Size

| Parameter Code | Explanation |
|---|---|
| CrPropertyStillImageTransSize_Original | Original |
| CrPropertyStillImageTransSize_SmallSize | Small Size JPEG/HEIF |

## CrDeviceProperty_RAW_J_PC_Save_Image

Get/Set the RAW+J PC Save Image

| Parameter Code | Explanation |
|---|---|
| CrPropertyRAWJPCSaveImage_RAWAndJPEG | RAW & JPEG |
| CrPropertyRAWJPCSaveImage_JPEGOnly | JPEG Only |
| CrPropertyRAWJPCSaveImage_RAWOnly | RAW Only |
| CrPropertyRAWJPCSaveImage_RAWAndHEIF | RAW & HEIF |
| CrPropertyRAWJPCSaveImage_HEIFOnly | HEIF Only |

## CrDeviceProperty_LiveView_Image_Quality

Get/Set the LiveView Quality

| Parameter Code | Explanation |
|---|---|
| CrPropertyLiveViewImageQuality_Low | Low |
| CrPropertyLiveViewImageQuality_High | High |

## CrDeviceProperty_CustomWB_Capture_Standby

Get the Custom WB Capture Standby Operation

| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBOperation_Disable | Disable |
| CrPropertyCustomWBOperation_Enable | Enable |

Execute the Custom WB Capture Standby

| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBCapture_Up | Up |
| CrPropertyCustomWBCapture_Down | Down |

## CrDeviceProperty_CustomWB_Capture_Standby_Cancel

Get the Custom WB Capture Standby Cancel Operation

| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBOperation_Disable | Disable |
| CrPropertyCustomWBOperation_Enable | Enable |

Execute the Custom WB Capture Standby Cancel

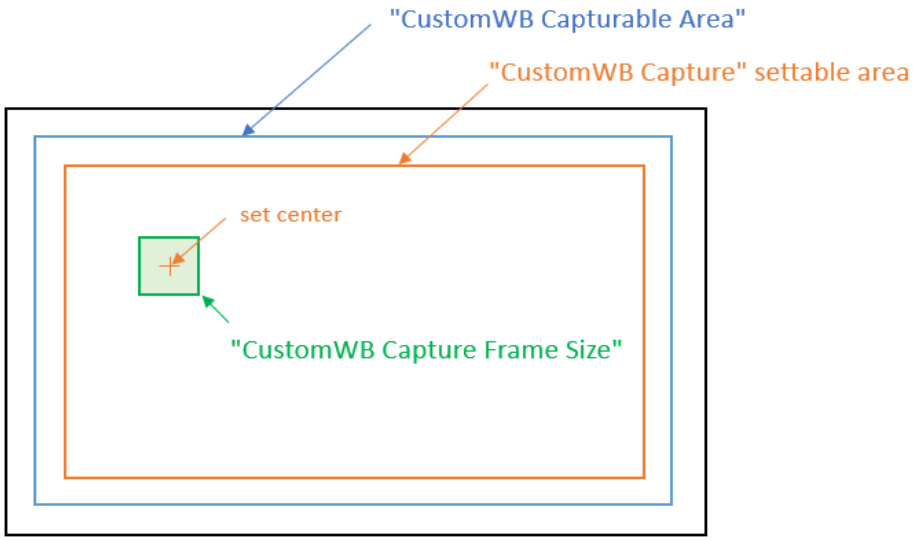
| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBCapture_Up | Up |
| CrPropertyCustomWBCapture_Down | Down |

## CrDeviceProperty_CustomWB_Capture

Execute the Custom WB Capture

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes<br>The enable range can be obtained from "Custom WB Capturable Area".<br>The settable area is more inside by half the Frame Size than “Custom WB Capturable Area".<br>Note:<br>The settable range varies depending on the model and aspect setting. |
| 0xFFFFFFFF | max | |
| 1 | step | |

Fig. Relationship between capture frame size and settable position



## CrDeviceProperty_SnapshotInfo

Get the Shooting File Info

| Value | Explanation | |
|---|---|---|
| 0x0000 | min | 0x0000:transferable file doesn't exit<br>0x0001-0x7FFF:exist file<br>If the value is over 0x8001(MSB is 0b01), it is possible to get the Shot files. |
| 0xFFFF | max | |
| 0x0001 | step | |

## CrDeviceProperty_BatteryRemain

Get the Battery Remaining (%)

| Value | Explanation |
|---|---|
| 0xFF(untaken) | min |
| 0x64(100%) | max |
| 0x01 | step |

## CrDeviceProperty_BatteryLevel

Get the Battery Level Indicator

| Parameter Code | Explanation |
|---|---|
| CrBatteryLevel_Fake | Fake Battery |
| CrBatteryLevel_PreEndBattery | Pre-End Battery |
| CrBatteryLevel_1_4 | Battery Level 1/4 |
| CrBatteryLevel_2_4 | Battery Level 2/4 |
| CrBatteryLevel_3_4 | Battery Level 3/4 |
| CrBatteryLevel_4_4 | Battery Level 4/4 |
| CrBatteryLevel_1_3 | Battery Level 1/3 |
| CrBatteryLevel_2_3 | Battery Level 2/3 |
| CrBatteryLevel_3_3 | Battery Level 3/3 |
| CrBatteryLevel_PreEnd_PowerSupply | Pre-End Battery with USB BusPower Supply |
| CrBatteryLevel_1_4_PowerSupply | Battery Level 1/4 with USB BusPower Supply |
| CrBatteryLevel_2_4_PowerSupply | Battery Level 2/4 with USB BusPower Supply |
| CrBatteryLevel_3_4_PowerSupply | Battery Level 3/4 with USB BusPower Supply |
| CrBatteryLevel_4_4_PowerSupply | Battery Level 4/4 with USB BusPower Supply |
| CrBatteryLevel_USBPowerSupply | USB BusPower Supply |

## CrDeviceProperty_RecordingState

Get the Movie Recording State

| Parameter Code | Explanation |
|---|---|
| CrMovie_Recording_State_Not_Recording | Not Recording |
| CrMovie_Recording_State_Recording | Recording |
| CrMovie_Recording_State_Recording_Failed | Recording Failed |
| CrMovie_Recording_State_IntervalRec_Waiting_Record | Waiting Record<br>(Time Lapse Movie recording) |

## CrDeviceProperty_LiveViewStatus

LiveView Status

| Parameter Code | Explanation |
|---|---|
| CrLiveView_Disable | LiveView Support but Disable just now :If this value is set, the host should not get the LiveView Image. |
| CrLiveView_Enable | LiveView Support and Enable :The host can get the LiveView Image and activate LiveView button if have. |
| CrLiveView_NotSupport | LiveView Not Support :Just definition, If the camera doesn't support Liveview, the host can't get this property by any operation. |

## CrDeviceProperty_FocusIndication

Get the Focus Indication

| Parameter Code | Explanation |
|---|---|
| CrFocusIndicator_Unlocked | Unlock |
| CrFocusIndicator_Focused_AF_S | [AF-S]Focussed, and AF Locked State |
| CrFocusIndicator_NotFocused_AF_S | [AF-S]Not focussed, and Low Contrast State |
| CrFocusIndicator_TrackingSubject_AF_C | [AF-C]Tracking Subject motion |
| CrFocusIndicator_Focused_AF_C | [AF-C]Focussed State |
| CrFocusIndicator_NotFocused_AF_C | [AF-C]Not focussed, and Low Contrast State |

## CrDeviceProperty_MediaSLOT1_Status

Get the Media (SLOT1) Status

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Parameter Code | Explanation |
|---|---|
| CrSlotStatus_OK | OK |
| CrSlotStatus_NoCard | No card |
| CrSlotStatus_CardError | Card error |
| CrSlotStatus_RecognizingOrLockedError | Card recognizing/Card locked and DB error |

## CrDeviceProperty_MediaSLOT1_RemainingNumber

Get the Remaining number shots of Media (SLOT1)

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | Unit is the remaining number of shots. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_MediaSLOT1_RemainingTime

Get the Remaining shooting time of Media (SLOT1)

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | Unit is second, the remaining time of movie recording. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_MediaSLOT1_FormatEnableStatus

Get the Media Full Format Enable Status(SLOT1)

| Parameter Code | Explanation |
|---|---|
| CrMediaFormat_Disable | Disable |
| CrMediaFormat_Enable | Enable |

## CrDeviceProperty_MediaSLOT2_Status

Get the Media (SLOT2) Status

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Parameter Code | Explanation |
|---|---|
| CrSlotStatus_OK | OK |
| CrSlotStatus_NoCard | No card |
| CrSlotStatus_CardError | Card error |
| CrSlotStatus_RecognizingOrLockedError | Card recognizing/Card locked and DB error |

## CrDeviceProperty_MediaSLOT2_RemainingNumber

Get the Remaining number shots of Media (SLOT2)

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | Unit is the remaining number of shots. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_MediaSLOT2_RemainingTime

Get the Remaining shooting time of Media (SLOT2)

Please set the function of camera “Rec. Media Settings: Recording Mode” to Standard when using ILCE-7RM4 and ILCE-7RM4A.

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | Unit is second, the remaining time of movie recording. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_MediaSLOT2_FormatEnableStatus

Get the Media Full Format Enable Status(SLOT2)

| Parameter Code | Explanation |
|---|---|
| CrMediaFormat_Disable | Disable |
| CrMediaFormat_Enable | Enable |

## CrDeviceProperty_Media_FormatProgressRate

Get the Media Format Progress Rate

| Value | Explanation |
|---|---|
| 0x00000000 | Invalid |
| Other than above values | Progress rate<br>Lower 16bit is denominator, Higher 16bit is molecules.<br>Calculate the progress rate each time.<br>e.g.) 0x003600C8 means 27%. (by the following calculations. (0x36/0xC8) * 100) |

## CrDeviceProperty_Interval_Rec_Status

Get the Interval REC Status

| Parameter Code | Explanation |
|---|---|
| CrIntervalRecStatus_WaitingStart | Waiting Start |
| CrIntervalRecStatus_IntervalShooting | Interval Shooting |

## CrDeviceProperty_CustomWB_Execution_State

Get the Custom WB Execution State

| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBExecutionState_Invalid | Invalid |
| CrPropertyCustomWBExecutionState_Standby | Standby |
| CrPropertyCustomWBExecutionState_Capturing | Capturing |
| CrPropertyCustomWBExecutionState_OperatingCamera | Operating Camera |

## CrDeviceProperty_CustomWB_Capturable_Area

Get the Custom WB Capturable Area(x,y)

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | The device can get the capturable area of Custom WB Capturing with this property.<br>The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes<br>This value varies depends on the model and aspect setting.<br>e.g.)<br>min 0x00200020 means TopLeft=32,32. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_CustomWB_Capture_Frame_Size

Get the Custom WB Capture Frame Size(x,y)

| Value | Explanation | |
|---|---|---|
| 0x00000000 | min | The frame width is set in the upper two bytes and the frame height is set in the lower two bytes<br>This value is currently 0x00400040 (64x64) fixed. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_CustomWB_Capture_Operation

Get the Custom WB Capture Operation Enable Status

| Parameter Code | Explanation |
|---|---|
| CrPropertyCustomWBOperation_Disable | Disable |
| CrPropertyCustomWBOperation_Enable | Enable |

## CrDeviceProperty_Zoom_Operation_Status

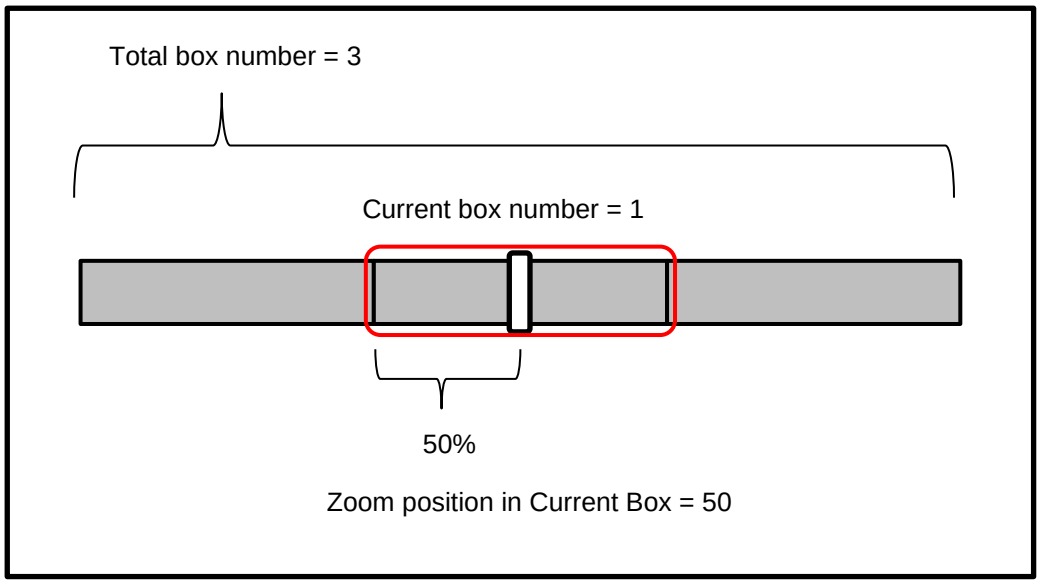
Get the Zoom Operation Enable Status

| Parameter Code | Explanation |
|---|---|
| CrZoomOperationEnableStatus_Disable | Disable |
| CrZoomOperationEnableStatus_Enable | Enable |

## CrDeviceProperty_Zoom_Bar_Information

Get the Zoom Bar Information

| Value | | Explanation |
|---|---|---|
| 31-24bit | | Total box number |
| | 0 | min |
| | 0xFF | max |
| | 1 | step |
| 23-16bit | | Current box number |
| | 0 | min |
| | 0xFF | max |
| | 1 | step |
| 15- 0bit | | Zoom position in Current Box |
| | 0x00 | min |
| | 0x64 | max |
| | 0x01 | step |



## CrDeviceProperty_Zoom_Type_Status

Get the Zoom Type Status

| Parameter Code | Explanation |
|---|---|
| CrZoomTypeStatus_OpticalZoom | Optical zoom only |
| CrZoomTypeStatus_SmartZoom | Smart zoom only |
| CrZoomTypeStatus_ClearImageZoom | Clear image zoom |
| CrZoomTypeStatus_DigitalZoom | Digital zoom |

## CrDeviceProperty_MediaSLOT1_FileType

Get/Set the File Format(Still) of media(SLOT1)

This property is effective when Recording Media for still images is set to "Sort Recording".
For ILCE-1 : MENU > Shooting > Media > Rec. Media Settings > Recording Media

This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrFileType_RawJpeg | RAW+JPEG |
| CrFileType_Jpeg | JPEG |
| CrFileType_Raw | RAW |
| CrFileType_RawHeif | RAW+HEIF |
| CrFileType_Heif | HEIF |

## CrDeviceProperty_MediaSLOT2_FileType

Get/Set the File Format(Still) of media(SLOT2)

This property is effective when Recording Media for still images is set to "Sort Recording".
For ILCE-1 : MENU > Shooting > Media > Rec. Media Settings > Recording Media

This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrFileType_RawJpeg | RAW+JPEG |
| CrFileType_Jpeg | JPEG |
| CrFileType_Raw | RAW |
| CrFileType_RawHeif | RAW+HEIF |
| CrFileType_Heif | HEIF |

## CrDeviceProperty_MediaSLOT1_JpegQuality

Get/Set the JPEG Quality of media(SLOT1)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT1_FileType" is set to "CrFileType_Jpeg" or "CrFileType_Heif".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > JPEG Quality/HEIF Quality
This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrJpegQuality_Light | Light |
| CrJpegQuality_Standard | Standard |
| CrJpegQuality_Fine | Fine |
| CrJpegQuality_ExFine | Extra fine |

## CrDeviceProperty_MediaSLOT2_JpegQuality

Get/Set the JPEG Quality of media(SLOT2)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT2_FileType" is set to "CrFileType_Jpeg" or "CrFileType_Heif".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > JPEG Quality/HEIF Quality
This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrJpegQuality_Light | Light |
| CrJpegQuality_Standard | Standard |
| CrJpegQuality_Fine | Fine |
| CrJpegQuality_ExFine | Extra fine |

## CrDeviceProperty_MediaSLOT1_ImageSize

Get/Set the Image Size of media(SLOT1)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT1_FileType" is set to "CrFileType_Jpeg" or "CrFileType_Heif".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > JPEG Image Size/HEIF Image Size
This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrImageSize_L | L |
| CrImageSize_M | M |
| CrImageSize_S | S |

## CrDeviceProperty_MediaSLOT2_ImageSize

Get/Set the Image Size of media(SLOT2)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT2_FileType" is set to "CrFileType_Jpeg" or "CrFileType_Heif".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > JPEG Image Size/HEIF Image Size
This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrImageSize_L | L |
| CrImageSize_M | M |
| CrImageSize_S | S |

## CrDeviceProperty_RAW_FileCompressionType

Get/Set the compression type of RAW file

This setting is related to “CrDeviceProperty_CompressionFileFormatStill”.

| Parameter Code | Explanation |
|---|---|
| CrRAWFile_Uncompression | Uncompression |
| CrRAWFile_Compression | Compression |
| CrRAWFile_LossLess | Lossless Compression |
| CrRAWFile_LossLessS | Lossless S |
| CrRAWFile_LossLessM | Lossless M |
| CrRAWFile_LossLessL | Lossless L |

## CrDeviceProperty_MediaSLOT1_RAW_FileCompressionType

Get/Set the compression type of RAW file in media(SLOT1)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT1_FileType" is set to "CrFileType_Raw".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > File Format/RAW File Type

| Parameter Code | Explanation |
|---|---|
| CrRAWFile_Uncompression | Uncompression |
| CrRAWFile_Compression | Compression |
| CrRAWFile_LossLess | Lossless Compression |
| CrRAWFile_LossLessS | Lossless S |
| CrRAWFile_LossLessM | Lossless M |
| CrRAWFile_LossLessL | Lossless L |

## CrDeviceProperty_MediaSLOT2_RAW_FileCompressionType

Get/Set the compression type of RAW file in media(SLOT2)

This property is effective when Recording Media for still images is set to "Sort Recording", and "CrDeviceProperty_MediaSLOT2_FileType" is set to "CrFileType_Raw".
For ILCE-1 : MENU > Shooting > Image Quality > Image Quality Settings > File Format/RAW File Type

| Parameter Code | Explanation |
|---|---|
| CrRAWFile_Uncompression | Uncompression |
| CrRAWFile_Compression | Compression |
| CrRAWFile_LossLess | Lossless Compression |
| CrRAWFile_LossLessS | Lossless S |
| CrRAWFile_LossLessM | Lossless M |
| CrRAWFile_LossLessL | Lossless L |

## CrDeviceProperty_MediaSLOT1_QuickFormatEnableStatus

Get the Media Quick Format Enable Status(SLOT1)

| Parameter Code | Explanation |
|---|---|
| CrMediaFormat_Disable | Disable |
| CrMediaFormat_Enable | Enable |

## CrDeviceProperty_MediaSLOT2_QuickFormatEnableStatus

Get the Media Quick Format Enable Status(SLOT2)

| Parameter Code | Explanation |
|---|---|
| CrMediaFormat_Disable | Disable |
| CrMediaFormat_Enable | Enable |

## CrDeviceProperty_Cancel_Media_FormatEnableStatus

Get the status of whether the media format is cancelable or not.

This property changes during Full formatting.

| Parameter Code | Explanation |
|---|---|
| CrCancelMediaFormat_Disable | Disable |
| CrCancelMediaFormat_Enable | Enable |

## CrDeviceProperty_ZoomAndFocusPosition_Save

Get/Set the Save Zoom&FocusPosition Preset.

| Parameter Code | Explanation |
|---|---|
| Variable | Save preset number<br><br>The current focus position, Optical Zoom position (Power Zoom lens only), and lens information are stored in the specified preset number.<br><br>With CrDeviceProperty_ZoomAndFocusPosition_Load, you can get the saved preset data and restore it to the same state.<br><br>Stored preset data will not be deleted even after initializing the camera. If you specify a preset number that is already in use, that preset number will be overwritten with the new preset data.<br><br>e.g.)<br><br>{0x00,0x01,0x02} means numbers 0 to 2 can be used |

## CrDeviceProperty_ZoomAndFocusPosition_Load

Get/Set the Load Zoom&FocusPosition Preset.

| Parameter Code | Explanation |
|---|---|
| Variable | Load preset number<br><br>Note:<br><br>If a lens other than the saved lens is attached, the focus / zoom position cannot be reproduced. In that case, it will notify you of CrWarning_ZoomAndFocusPosition_DifferentLens.<br><br>Environmental changes or the focus position of the lens, such as Near/Far edge vicinity, may cause errors in the original position the lens returns.<br>Please use this property with larger Aperture Value (F-Number) to deepen the depth of field and confirm the focus position the lens returns in advanced. |

## CrDeviceProperty_Remocon_Zoom_Speed_Type

Get/Set the Remocon Zoom Speed Type.

| Parameter Code | Explanation |
|---|---|
| CrRemoconZoomSpeedType_Invalid | Invalid |
| CrRemoconZoomSpeedType_Variable | Variable<br><br>Related to CrDeviceProperty_Zoom_Operation and CrDeviceProperty_Zoom_Speed_Range. |
| CrRemoconZoomSpeedType_Fixed | Fixed |

## CrDeviceProperty_Zoom_Speed_Range

Get the Zoom Speed Range.

| Parameter Code | Explanation | |
|---|---|---|
| Variable<br>(Negative number) | min | Zoom in speed is a positive number and zoom out speed is a negative number.<br>Valid when CrDeviceProperty_Remocon_Zoom_Speed_Type is CrRemoconZoomSpeedType_Variable.<br><br>Note:<br>The actual zoom speed depends on the specifications of each lens model. |
| Variable<br>(Positive number) | max | |
| 1 | step | |

## CrDeviceProperty_SdkControlMode

Get the Sdk Control Mode.

| Parameter Code | Explanation |
|---|---|
| CrSdkControlMode_Remote | Remote Control Mode<br><br>The default mode when connected to the camera. This mode is for shooting remotely. It is possible to change device properties for shooting such as shutter speed and ISO value.<br>If you do not specify openMode of the connect function, connect in this mode. |
| CrSdkControlMode_ContentsTransfer | Contents Transfer Mode<br><br>This mode is for pulling out the contents of the media inserted in the camera slot. |

See “Supporting physical layer” for models that support each mode.

## CrDeviceProperty_ContentsTransferStatus

Get the content transfer status

| Parameter Code | Explanation |
|---|---|
| CrContentsTransfer_OFF | OFF<br><br>The state in which the camera cannot transfer content |
| CrContentsTransfer_ON | ON |

## CrDeviceProperty_ContentsTransferCancelEnableStatus

Get the cancelability status of content transfer.

| Parameter Code | Explanation |
|---|---|
| CrCancelContentsTransfer_Disable | Disable |
| CrCancelContentsTransfer_Enable | Enable |

## CrDeviceProperty_ContentsTransferProgress

Gets the handle and progress of the content during transfer

| Parameter Code | Explanation |
|---|---|
| 63-32bit | CrContentHandle<br><br>Content handle during transfer processing |
| 31-0bit | 0-100<br><br>Transfer progress rate. Unit is percent(%)<br>Content with a large file size is acquired in multiple steps.<br>The acquisition time changes depending on the size of the file size.<br>With this progress rate, you can predict that the transfer of the specified content will be completed. |

## CrDeviceProperty_IrisModeSetting

Get/Set the Iris Mode Setting

In ILC, enabled when "CrDeviceProperty_ExposureCtrlType" is in "Flexible Exposure Mode".

| Parameter Code | Explanation |
|---|---|
| CrIrisMode_Automatic | Automatic |
| CrIrisMode_Manual | Manual |

## CrDeviceProperty_ShutterModeSetting

Get/Set the Shutter Mode Setting

In ILC, enabled when "CrDeviceProperty_ExposureCtrlType" is in "Flexible Exposure Mode".

| Parameter Code | Explanation |
|---|---|
| CrShutterMode_Automatic | Automatic |
| CrShutterMode_Manual | Manual |

## CrDeviceProperty_GainControlSetting

Get/Set the Gain Control Setting

| Parameter Code | Explanation |
|---|---|
| CrGainControl_Automatic | Automatic |
| CrGainControl_Manual | Manual |

## CrDeviceProperty_GainBaseIsoSensitivity

Get/Set the Gain Base ISO Sensitivity

| Parameter Code | Explanation |
|---|---|
| CrGainBaseIsoSensitivity_High | High Level |
| CrGainBaseIsoSensitivity_Low | Low Level |

## CrDeviceProperty_GainBaseSensitivity

Get/Set the Gain Base Sensitivity

| Parameter Code | Explanation |
|---|---|
| CrGainBaseSensitivity_High | High Level |
| CrGainBaseSensitivity_Low | Low Level |

## CrDeviceProperty_ExposureIndex

Get/Set the Exposure Index

| Parameter Code | Explanation |
|---|---|
| Variable | Exposure Index<br><br>Set the EI value, The set value varies depending on the model and the setting status of the camera.<br>See GetDisplayStringList() for display character string and highlight latitude list associated with EI.<br><br>Ex.) If setting with "200EI / 4.0E", set 0x00C8. |

## CrDeviceProperty_BaseLookValue

Get/Set the BaseLook Value

<table>
<tr><th>Parameter Code</th><th colspan="2">Explanation</th></tr>
<tr><td>15-8bit<br><br>CrBaseLookValue_Preset(0x00)<br>CrBaseLookValue_User(0x01)</td><td>Kind</td><td rowspan="2">16bit value that combines Kind(upper 8bit) and Index (lower 8bit)<br><br>Ex.)<br>0x0003 = 3(Preset)<br>0x0108 = 8(User)<br><br>It may increase or decrease because it varies depending on the model and setting status.<br>See the GetDisplayStringList() for display character string.</td></tr>
<tr><td>0-7bit</td><td>Index</td></tr>
</table>

## CrDeviceProperty_PlaybackMedia

Get/Set the Playback Media

| Parameter Code | Explanation |
|---|---|
| CrPlaybackMedia_Slot1 | SLOT1 |
| CrPlaybackMedia_Slot2 | SLOT2 |

## CrDeviceProperty_DispModeCandidate

Get the Monitor DISP(Screen Display) Mode Candidate

| Parameter Code | | Explanation |
|---|---|---|
| GetCurrentValue() is always zero.<br>In GetValues(), one or more of the following items (Bit positions) that can be set in SetCurrentValue() of CrDeviceProperty_DispModeSetting are set. | | |
| | CrDispModeBitNum_GraphicDisplay | Graphic Display |
| | CrDispModeBitNum_DisplayAllInfo | Display All Information |
| | CrDispModeBitNum_Histogram | Histogram |
| | CrDispModeBitNum_Level | Level |
| | CrDispModeBitNum_NoDispInfo | No Display Information |
| | CrDispModeBitNum_NoDispInfoExposureOn | No Display Information Exposure:On |
| | CrDispModeBitNum_NoDispInfoExposureTimeOut | No Display Information Timeout |
| | CrDispModeBitNum_ForViewFinder | For Viewfinder |
| | CrDispModeBitNum_MonitorOff | Monitor Off |
| | | ex) If the camera supports Display All Information, Histogram, Level, No Display Information, GetValues() will be set to the following four values.<br><br>values[0] = 0x00000002 (Display All Information)<br>values[1] = 0x00000004 (Histogram)<br>values[2] = 0x00000008 (Level)<br>values[3] = 0x00000010 (No Display Information) |

## CrDeviceProperty_DispModeSetting

Get/Set the Monitor DISP(Screen Display) Mode Setting

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Set whether to enable or disable selectable items in CrDeviceProperty_DispMode.<br>The only candidates that can be selected in CrDeviceProperty_DispMode are the items(bit position) that are set to enable(turn on the bit) in this property.<br><br>Note:<br>Not all items can be disabled. Be sure to set one or more items(bit position) to enable(turn on the bit).<br>Refer to "About the Monitor DISP(Screen Display) for camera body". |
| Variable | max | |
| 1 | step | |

## CrDeviceProperty_DispMode

Get/Set the Monitor DISP(Screen Display) Mode

You can select one of the items enabled in CrDeviceProperty_DispModeSetting.

| Parameter Code | Explanation |
|---|---|
| CrDispMode_GraphicDisplay | Graphic Display |
| CrDispMode_DisplayAllInfo | Display All Information |
| CrDispMode_NoDispInfo | No Display Information |
| CrDispMode_Histogram | Histogram |
| CrDispMode_Level | Level |
| CrDispMode_ForViewFinder | For Viewfinder |
| CrDispMode_MonitorOff | Monitor Off |

## CrDeviceProperty_TouchOperation

Get/Set the Touch Operation Setting

| Parameter Code | Explanation |
|---|---|
| CrTouchOperation_Off | Off |
| CrTouchOperation_On | On |
| CrTouchOperation_PlaybackOnly | On: Playback only |

## CrDeviceProperty_SelectFinder

Get/Set the Finder/Monitor Setting

| Parameter Code | Explanation |
|---|---|
| CrSelectFinder_Auto | Auto |
| CrSelectFinder_ViewFinder_M | Viewfinder(Manual) |
| CrSelectFinder_Monitor_M | Monitor(Manual) |

## CrDeviceProperty_AutoPowerOffTemperature

Get/Set the Auto Power OFF Temperature

| Parameter Code | Explanation |
|---|---|
| CrAutoPowerOffTemperature_Standard | Standard |
| CrAutoPowerOffTemperature_High | High |

## CrDeviceProperty_BodyKeyLock

Get/Set the Body Key Lock

| Parameter Code | Explanation |
|---|---|
| CrBodyKey_Unlock | Unlock |
| CrBodyKey_Lock | Lock |

## CrDeviceProperty_ImageID_Num_Setting

Get/Set the Image ID(Numerical) Setting

See "GPS information and shooting image link" in Tips / Trouble shooting for how to use it.

| Parameter Code | Explanation |
|---|---|
| CrImageIDNumSetting_Off | OFF<br>Do not save the CurrentValue of CrDeviceProperty_ImageID_Num to the Exif tag of the image. |
| CrImageIDNumSetting_On | ON<br>Save the CurrentValue of CrDeviceProperty_ImageID_Num to the Exif tag of the image.<br><br>Caution:<br>When the power of the camera is turned off or the “PC Remote” is “Off”, it is initialized to OFF. |

## CrDeviceProperty_ImageID_Num

Get/Set the Image ID(Numerical Value)

See "GPS information and shooting image link" in Tips / Trouble shooting for how to use it.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | By specifying a value in this property before shooting, the value specified in the Exif tag of the image file shot after that will be saved.<br>Save the value in the Exif tag of the image file only if CrDeviceProperty_ImageID_Num_Setting is CrImageIDNumSetting_On.<br><br>If you shoot immediately after setting, it may not be recorded in Exif. Be sure to Get and make sure that the set value and the Get value match before shooting.<br><br>Note:<br>The Exif tag for Image ID (Numerical Value) is 0x2042. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ImageID_String

Get/Set the Image ID(String)

See "GPS information and shooting image link" in Tips / Trouble shooting for how to use it.

| Parameter Code | Explanation |
|---|---|
| String | By specifying a value in this property before shooting, the value specified in the Exif tag of the image file shot after that will be saved. You can save up to 64 characters(128byte with UTF16BE). If you set a size larger than that, it will not be saved.<br>If blank ("") is set, Exif tags are not save in the image.<br><br>Note:<br>The Exif tag for Image ID (String) is 0x2043. |

## CrDeviceProperty_ExposureCtrlType

Get/Set the Exposure Control Type

| Parameter Code | Explanation |
|---|---|
| CrExposureCtrlType_PASMMode | P/A/S/M Mode |
| CrExposureCtrlType_FlexibleExposureMode | Flexible Exposure Mode |

## CrDeviceProperty_MonitorLUTSetting

Get/Set the Monitor LUT Setting

| Parameter Code | Explanation |
|---|---|
| CrMonitorLUT_OFF | OFF |
| CrMonitorLUT_ON | ON |

## CrDeviceProperty_IsoCurrentSensitivity

Get the ISO Current Sensitivity

| Value | Explanation |
|---|---|
| - | value : bit 28-31 extension, bit 24-27 ISO mode , bit 0-23 ISO value.<br><br>Real ISO value : when bits 0-23 are other than CrISO_AUTO(0xFFFFFF).<br><br>e.g.) 0x00000140 = 320 |

## CrDeviceProperty_CameraSetting_SaveOperationEnableStatus

Get the Camera-Setting Save Operation Enable Status

DownloadSettingFile() is possible when this property is Enable.

| Parameter Code | Explanation |
|---|---|
| CrCameraSettingSaveOperation_Disable | Disable |
| CrCameraSettingSaveOperation_Enable | Enable |

## CrDeviceProperty_CameraSetting_ReadOperationEnableStatus

Get the Camera-Setting Read Operation Enable Status

UploadSettingFile() is possible when this property is Enable.

| Parameter Code | Explanation |
|---|---|
| CrCameraSettingReadOperation_Disable | Disable |
| CrCameraSettingReadOperation_Enable | Enable |

## CrDeviceProperty_CameraSetting_SaveRead_State

Get the Camera-Setting Save/Read State

| Parameter Code | Explanation |
|---|---|
| CrCameraSettingSaveReadState_Idle | Idle |
| CrCameraSettingSaveReadState_Reading | Reading |

## CrDeviceProperty_CameraSettingsResetEnableStatus

Get the Camera Setting Reset Enable State

| Parameter Code | Explanation |
|---|---|
| CrCameraSettingsReset_Disable | Disable |
| CrCameraSettingsReset_Enable | Enable |

## CrDeviceProperty_APS_C_or_Full_SwitchingSetting

Get the APS-C or Full Switching Setting

| Parameter Code | Explanation |
|---|---|
| CrAPS_C_or_Full_SwitchingSetting_Full | Full |
| CrAPS_C_or_Full_SwitchingSetting_APS_C | APS-C |

## CrDeviceProperty_APS_C_or_Full_SwitchingEnableStatus

Get the APS-C or Full Switching Status

| Parameter Code | Explanation |
|---|---|
| CrAPS_C_or_Full_Switching_Disable | Disable |
| CrAPS_C_or_Full_Switching_Enable | Enable |

## CrDeviceProperty_FocalDistanceInMeter

Get/Set the Focal Distance in Meter

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | 1000 times the real value of focal distance in meters.<br>If current value is CrFocalDistance_Infinity(0xFFFFFFFF), ∞.<br><br>e.g.) 0x00005014 = 20500 /1000 = 20.5 meter<br>e.g.) 0x00030D40 = 200000 /1000 = 200 meter |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_FocalDistanceInFeet

Get/Set the Focal Distance in Feet

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | 1000 times the real value of focal distance in feet.<br>If current value is CrFocalDistance_Infinity(0xFFFFFFFF), ∞.<br><br>e.g.) 0x00005014 = 20500 /1000 = 20.5 feet<br>e.g.) 0x00030D40 = 200000 /1000 = 200 feet |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_FocalDistanceUnitSetting

Get/Set the Focal Distance Unit Setting

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrFocalDistanceUnitSetting_Meter | Meter |
| CrFocalDistanceUnitSetting_Feet | Feet |

## CrDeviceProperty_DigitalZoomScale

Get/Set the Digital Zoom Scale

For ILME-FX6, only gets are supported. refs Zoom Operation / Zoom Scale.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | 1000 times the real value of zoom scale.<br>The resolution of the CurrentValue is the step value.<br>The CurrentValue increases or decreases with each step value.<br><br>Ex.) 0x000004B0 = 1200 /1000 = x1.2<br><br>CrDeviceProperty_Zoom_Scale shows the total scale of digital and optical. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ZoomDistance

Get/Set the Zoom Distance

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Units of 0.001mm.<br>min/max/CurrentValue should be set in units of “step”.<br><br>Ex.) min: 18000, max: 55000, step: 1000, value: 20000<br>(min = 18mm, max = 55mm, value = 20mm)<br><br>The maximum value as a protocol is 4294967 mm.<br><br>Note:<br>Indicates the distance when CrDeviceProperty_ZoomDistanceUnitSetting is CrZoomDistanceUnitSetting_mm. When CrDeviceProperty_ZoomDistanceUnitSetting is CrZoomDistanceUnitSetting_percent, refer to CrDeviceProperty_Zoom_Bar_Information. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ZoomDistanceUnitSetting

Get/Set the Zoom Distance Unit Setting

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrZoomDistanceUnitSetting_mm | mm |
| CrZoomDistanceUnitSetting_percent | percent |

## CrDeviceProperty_ShutterModeStatus

Get/Set the Shutter Mode Status

| Parameter Code | Explanation |
|---|---|
| CrShutterModeStatus_Off | OFF |
| CrShutterModeStatus_Speed | Speed |
| CrShutterModeStatus_Angle | Angle |
| CrShutterModeStatus_ECS | ECS |
| CrShutterModeStatus_Auto | Auto |

## CrDeviceProperty_ShutterSlow

Get/Set the Shutter Slow

| Parameter Code | Explanation |
|---|---|
| CrShutterSlow_Off | OFF |
| CrShutterSlow_On | ON |

## CrDeviceProperty_ShutterSlowFrames

Get/Set the Shutter Slow Frames

| Parameter Code | Explanation |
|---|---|
| CrShutterSlowFrames_Disable | - |
| Other than above values | Shutter Slow Frames Value |

## CrDeviceProperty_Movie_Recording_ResolutionForMain

Get/Set the Recording Resolution For Main(Movie)

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00000000<br>~ 0xFFFFFFFF) | Recording resolution(Width , Height)<br><br>The "Width" is set in the upper two bytes and the "Height" is set in the lower two bytes<br>if resolution (Width) is 1920, (Height) is 1080, set 0x07800438.<br>0x0780 = 0d1920, 0x0438 = 0d1080 |

## CrDeviceProperty_Movie_Recording_ResolutionForProxy

Get/Set the Recording Resolution For Proxy(Movie)

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00000000<br>~ 0xFFFFFFFF) | Recording resolution(Width , Height)<br><br>The "Width" is set in the upper two bytes and the "Height" is set in the lower two bytes<br>if resolution (Width) is 1920, (Height) is 1080, set 0x07800438.<br>0x0780 = 0d1920, 0x0438 = 0d1080 |

## CrDeviceProperty_Movie_Recording_FrameRateProxySetting

Get/Set the Recording Frame Rate Proxy Setting(Movie)

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrRecordingFrameRateSettingMovie_50p | 50p |
| CrRecordingFrameRateSettingMovie_25p | 25p |
| CrRecordingFrameRateSettingMovie_24p | 24p<br>Actual field frequency might be 23.98 except ILME-FX6. |
| CrRecordingFrameRateSettingMovie_23_98p | 23.98p |
| CrRecordingFrameRateSettingMovie_29_97p | 29.97p |
| CrRecordingFrameRateSettingMovie_59_94p | 59.94p |
| CrRecordingFrameRateSettingMovie_24_00p | 24.00p |
| CrRecordingFrameRateSettingMovie_119_88p | 119.88p |

See Table fr-1/2 in CrDeviceProperty_Movie_Recording_FrameRateSetting

## CrDeviceProperty_MovieShootingMode

Get/Set the Movie Shooting Mode

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrMovieShootingMode_Off | OFF |
| CrMovieShootingMode_CineEI | CineEI |
| CrMovieShootingMode_CineEIQuick | CineEI Quick |
| CrMovieShootingMode_Custom | Custom |
| CrMovieShootingMode_FlexibleISO | Flexible ISO |

## CrDeviceProperty_MovieShootingModeColorGamut

Get/Set the Movie Shooting Mode Color Gamut

See “Get the menu display string” for menu display characters.

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrMovieShootingModeColorGamut_S_Gamut3_Cine | S-Gamut3.Cine |
| CrMovieShootingModeColorGamut_S_Gamut3 | S-Gamut3 |

## CrDeviceProperty_MovieShootingModeTargetDisplay

Get/Set the Movie Shooting Mode Target Display

See “Get the menu display string” for menu display characters.

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrMovieShootingModeTargetDisplay_BT709 | BT.709 |
| CrMovieShootingModeTargetDisplay_BT2020 | BT.2020 |

## CrDeviceProperty_DepthOfFieldAdjustmentMode

Get/Set the Depth of Field Adjustment Mode

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrDepthOfFieldAdjustmentMode_OFF | OFF |
| CrDepthOfFieldAdjustmentMode_ON | ON |

## CrDeviceProperty_DepthOfFieldAdjustmentInterlockingMode

Get the Depth of Field Adjustment Interlocking Mode State

| Parameter Code | Explanation |
|---|---|
| CrDepthOfFieldAdjustmentInterlockingMode_NDInterlockingMode | ND interlocking mode |
| CrDepthOfFieldAdjustmentInterlockingMode_GainInterlockingMode | Gain interlocking mode |

## CrDeviceProperty_ColortempStep

Set the Color Temperature

Manipulating this device property updates the CurrentValue of CrDeviceProperty_Colortemp.

| Parameter Code | Explanation | |
|---|---|---|
| -30 | min | The CurrentValue of this device property is always zero.<br>This device property is used to update CrDeviceProperty_Colortemp.<br><br>The step value of this device property is synchronized with the step value of CrDeviceProperty_Colortemp, and if the step value of CrDeviceProperty_Colortemp is 100, updating to -1 using this device property will decrement the CurrentValue of CrDeviceProperty_Colortemp by 100.<br>And if you use this device property to update to +2, the CurrentValue of CrDeviceProperty_Colortemp will increase by 200. |
| 30 | max | |
| 1 | step | |

## CrDeviceProperty_WhiteBalanceModeSetting

Get/Set the White Balance Mode Setting

| Parameter Code | Explanation |
|---|---|
| CrWhiteBalanceModeSetting_Automatic | Automatic |
| CrWhiteBalanceModeSetting_Manual | Manual |

## CrDeviceProperty_WhiteBalanceTint

Get/Set the White Balance Tint

This device property can also be updated by CrDeviceProperty_WhiteBalanceTintStep.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | White Balance Tint<A/B> setting value, A and B can be switched by SW to control the CurrentValue remotely.<br><br>Note:<br>In ILME-FX6, it is always GetOnly, regardless of the return value of IsSetEnableCurrentValue(). |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_WhiteBalanceTintStep

Set the White Balance Tint

Manipulating this device property updates the CurrentValue of CrDeviceProperty_WhiteBalanceTint.

| Parameter Code | Explanation | |
|---|---|---|
| -198 | min | The CurrentValue of this device property is always zero.<br>This device property is used to update CrDeviceProperty_WhiteBalanceTint. |
| 198 | max | |
| 1 | step | |

## CrDeviceProperty_Focus_Operation

Execute the Focus Operation

This device property is valid when CrDeviceProperty_FocalDistanceInMeter or CrDeviceProperty_FocalDistanceInFeet is enabled.

| Parameter Code | Explanation |
|---|---|
| - | The CurrentValue of this device property is always zero. Update only.<br>Can be set within the range of CrDeviceProperty_Focus_Speed_Range.<br><br>Ex.) SetValue = 1 : Tele focus (focus speed=1)<br>SetValue = -3 : Wide focus (focus speed=3)<br>SetValue = 0 : Stop focus |

## CrDeviceProperty_Focus_Speed_Range

Get the Focus Speed Range

| Parameter Code | Explanation | |
|---|---|---|
| Variable<br>(Negative number) | min | A value that can be used for Focus Operation.<br>For example, when min is -5 and max is +5, it means that the focus drive speed can be specified in 5 steps.<br>The higher the number, the faster the focus drive speed.<br>The CurrentValue of this device property is always zero. |
| Variable<br>(Positive number) | max | |
| Variable | step | |

## CrDeviceProperty_ShutterECSSetting

Get/Set the Shutter ECS Setting

| Parameter Code | Explanation |
|---|---|
| CrShutterECSSetting_OFF | OFF |
| CrShutterECSSetting_ON | ON |

## CrDeviceProperty_ShutterECSNumber

Get/Set the Shutter ECS Number

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | This device property is used to specify Shutter ECS with a certain range of Index values. The upper and lower limits of the relative value operation by CrDeviceProperty_ShutterECSNumberStep are obtained with this device property.<br>If CrDeviceProperty_ShutterECSNumberStep is operated while the CurrentValue is min or max, the CurrentValue will not be changed.<br><br>Note:<br>In ILME-FX6, it is always GetOnly, regardless of the return value of IsSetEnableCurrentValue(). |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ShutterECSNumberStep

Set the Shutter ECS Number Step

| Parameter Code | Explanation | |
|---|---|---|
| -32768 | min | The CurrentValue of this device property is always zero.<br><br>Updating this property will be reflected in CrDeviceProperty_ShutterECSNumber. |
| 32767 | max | |
| 1 | step | |

## CrDeviceProperty_ShutterECSFrequency

Get/Set the Shutter ECS Frequency

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | 1000 times the real value of Shutter ECS Frequency |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_ButtonAssignmentAssignable1

Get/Set the Button Assignment Assignable 1

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | Assign a certain function to CrDeviceProperty_AssignableButton1 so that the function can be executed by button operation.<br><br>GetValues() contains a list of function-code that can be assigned to CrDeviceProperty_AssignableButton1. Function-code are 8-bit values, and the number (number of functions) varies depending on the model and setting status. You can use GetDisplayStringList() to get a list of assignable function names.<br><br>See “Get the menu display string” |

## CrDeviceProperty_ButtonAssignmentAssignable2

Get/Set the Button Assignment Assignable 2

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable3

Get/Set the Button Assignment Assignable 3

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable4

Get/Set the Button Assignment Assignable 4

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable5

Get/Set the Button Assignment Assignable 5

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable6

Get/Set the Button Assignment Assignable 6

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable7

Get/Set the Button Assignment Assignable 7

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable8

Get/Set the Button Assignment Assignable 8

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentAssignable9

Get/Set the Button Assignment Assignable 9

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_ButtonAssignmentLensAssignable1

Get/Set the Button Assignment LensAssignable 1

| Parameter Code | Explanation |
|---|---|
| Variable<br>(0x00 ~ 0xFF) | The specifications of this device property are the same as CrDeviceProperty_ButtonAssignmentAssignable1. |

## CrDeviceProperty_AssignableButton1

Get/Set the Assignable Button 1

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | Be sure to specify "Up" after specifying "Down". |
| CrAssignableButton_Down | Specify "Down" and execute the function assigned to CrDeviceProperty_ButtonAssignmentAssignable1.<br><br>It stays in the Down state (hold down the button) until CrAssignableButton_Up is set. |

## CrDeviceProperty_AssignableButton2

Get/Set the Assignable Button 2

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton3

Get/Set the Assignable Button 3

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton4

Get/Set the Assignable Button 4

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton5

Get/Set the Assignable Button 5

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton6

Get/Set the Assignable Button 6

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton7

Get/Set the Assignable Button 7

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton8

Get/Set the Assignable Button 8

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_AssignableButton9

Get/Set the Assignable Button 9

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_LensAssignableButton1

Get/Set the LensAssignable Button 1

| Parameter Code | Explanation |
|---|---|
| CrAssignableButton_Up | The specifications of this device property are the same as CrDeviceProperty_AssignableButton1. |
| CrAssignableButton_Down | |

## CrDeviceProperty_FocusModeSetting

Get/Set the Focus Mode Setting

| Parameter Code | Explanation |
|---|---|
| CrFocusModeSetting_Automatic | Automatic |
| CrFocusModeSetting_Manual | Manual |

## CrDeviceProperty_ShutterAngle

Get/Set the Shutter Angle

| Parameter Code | Explanation |
|---|---|
| CrShutterAngle_Disable | - |
| Other than above values | 1000 times the real value of Shutter Angle<br>e.g.) 0x0002BF20 = 180000 /1000 = 180<br>e.g.) 0x00015F90 = 90000 /1000 = 90 |

## CrDeviceProperty_ShutterSetting

Get/Set the Shutter Setting

| Parameter Code | Explanation |
|---|---|
| CrShutterSetting_OFF | OFF |
| CrShutterSetting_ON | ON |

## CrDeviceProperty_ShutterMode

Get/Set the Shutter Mode

| Parameter Code | Explanation |
|---|---|
| CrShutterMode_Speed | Speed |
| CrShutterMode_Angle | Angle |

## CrDeviceProperty_ShutterSpeedValue

Get/Set the Shutter Speed Value

| Parameter Code | Explanation |
|---|---|
| Variable | Upper four bytes: numerator, Lower four bytes: denominator |

## CrDeviceProperty_ShutterSpeedCurrentValue

Get the Shutter Speed Current Value

| Parameter Code | Explanation |
|---|---|
| Variable | Upper four bytes: numerator, Lower four bytes: denominator |

## CrDeviceProperty_NDFilter

Get/Set the ND Filter

| Parameter Code | Explanation |
|---|---|
| CrNDFilter_OFF | OFF |
| CrNDFilter_ON | ON |

## CrDeviceProperty_NDFilterMode

Get the ND Filter Mode

| Parameter Code | Explanation |
|---|---|
| CrNDFilterMode_Auto | Auto |
| CrNDFilterMode_Preset | Preset |
| CrNDFilterMode_PresetClear | Preset clear |
| CrNDFilterMode_Variable | Variable |
| CrNDFilterMode_VariableClear | Variable clear |

## CrDeviceProperty_NDFilterModeSetting

Get/Set the ND Filter Mode Setting

| Parameter Code | Explanation |
|---|---|
| CrNDFilterModeSetting_Automatic | Automatic |
| CrNDFilterModeSetting_Manual | Manual |

## CrDeviceProperty_NDFilterValue

Get/Set the ND Filter Value

| Parameter Code | Explanation |
|---|---|
| CrNDFilterValue_Nothing | nothing to display. |
| Other than above values | The real value of ND Filter (Upper four bytes: numerator, Lower four bytes: denominator) |

## CrDeviceProperty_GainUnitSetting

Get/Set the Gain Unit Setting

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrGainUnitSetting_dB | dB |
| CrGainUnitSetting_ISO | ISO |

## CrDeviceProperty_GaindBValue

Get/Set the Gain dB Value

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Gain dB value. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_GaindBCurrentValue

Get the Gain dB Current Value

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Current value when Gain dB auto. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_AWB

Get/Set the AWB

| Parameter Code | Explanation |
|---|---|
| CrAWB_Up | Up |
| CrAWB_Down | Down |

## CrDeviceProperty_SceneFileIndex

Get/Set the SceneFile Index

| Parameter Code | Explanation |
|---|---|
| - | It may increase or decrease because it varies depending on the model and setting status.<br>Get the display character and list of value with GetDisplayStringList(). |

## CrDeviceProperty_CurrentSceneFileEdited

Get the Current SceneFile Edited Info.

| Parameter Code | Explanation |
|---|---|
| CrCurrentSceneFileEdited_Unedited | Unedited |
| CrCurrentSceneFileEdited_Edited | Edited |

## CrDeviceProperty_MoviePlayButton

Get/Set the Movie Play button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Be sure to specify "Up" after specifying "Down". |
| CrMovieXButton_Down | Specify "Down" when you start movie play. |

## CrDeviceProperty_MoviePlayPauseButton

Get/Set the Movie Play Pause button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Pause movie playback.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MoviePlayStopButton

Get/Set the Movie Play Stop button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Stop movie playback.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MovieForwardButton

Get/Set the Movie Forward button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Fast-forward playback of movie.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MovieRewindButton

Get/Set the Movie Rewind button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Rewind playback of movie.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MovieNextButton

Get/Set the Movie Next button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Moves to the top of the next movie.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MoviePrevButton

Get/Set the Movie Prev button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Moves to the top of the previous movie.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_MovieRecReviewButton

Get/Set the Movie RecReview button

| Parameter Code | Explanation |
|---|---|
| CrMovieXButton_Up | Play the last recorded movie file.<br>The specifications of this device property are the same as CrDeviceProperty_MoviePlayButton. |
| CrMovieXButton_Down | |

## CrDeviceProperty_FaceEyeDetectionAF

Get/Set Face Eye Detection AF

| Parameter Code | Explanation |
|---|---|
| CrFaceEyeDetectionAF_Off | Off |
| CrFaceEyeDetectionAF_FaceEyeOnlyAF | Face/Eye Only AF |
| CrFaceEyeDetectionAF_FaceEyePriorityAF | Face/Eye Priority AF |

## CrDeviceProperty_AFTransitionSpeed

Get/Set AF Transition speed

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Note:<br>The range value may change depending on the model. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_AFSubjShiftSens

Get/Set AF Subj Shift Sens

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Note:<br>The range value may change depending on the model. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_AFAssist

Get/Set the AF Assist

| Parameter Code | Explanation |
|---|---|
| CrAFAssist_Off | OFF |
| CrAFAssist_On | ON |

## CrDeviceProperty_NDPresetOrVariableSwitchingSetting

Get/Set the ND PRESET or VARIABLE Switching Setting

| Parameter Code | Explanation |
|---|---|
| CrNDPresetOrVariableSwitchingSetting_Preset | PRESET |
| CrNDPresetOrVariableSwitchingSetting_Variable | VARIABLE |

## CrDeviceProperty_FunctionOfRemoteTouchOperation

Get/Set the Function of Remote Touch Operation

| Parameter Code | Explanation |
|---|---|
| CrFunctionOfRemoteTouchOperation_Tracking_AF | Tracking AF |
| CrFunctionOfRemoteTouchOperation_Spot_AF | Spot AF |
| CrFunctionOfRemoteTouchOperation_AFAreaSelect | AF Area Select |

## CrDeviceProperty_RemoteTouchOperation

Execute Remote Touch Operation(x,y)

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | The CurrentValue of this property is always zero.<br>This property can only be executed if CrDeviceProperty_RemoteTouchOperationEnableStatus is Enable.<br><br>The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes<br>The range of X is 0 ~ 639 (0x027F), and the range of Y is 0 ~ 479 (0x01DF).<br><br>Note:<br>For ILCE-7SM3 and ILCE-7C, only available in movie mode. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_MoviePlayingState

Get the Movie Playing State

| Parameter Code | Explanation |
|---|---|
| CrMoviePlayingState_NotPlaying | Not Playing |
| CrMoviePlayingState_Playing | Playing |

## CrDeviceProperty_MoviePlayingSpeed

Get Movie Playing Speed

| Parameter Code | Explanation |
|---|---|
| CrMoviePlayingSpeed_Nothing | nothing to display. |
| Other than above values | The real value of Clip Playing Speed (Upper four bytes: numerator, Lower four bytes: denominator)<br><br>The numerator is int32_t type and the denominator is uint32_t type. |

## CrDeviceProperty_MediaSLOT1Player

Get the Media SLOT1 Player

| Parameter Code | Explanation |
|---|---|
| CrMediaPlayer_None | None |
| CrMediaPlayer_Player | Player |
| CrMediaPlayer_Recorder | Recorder |
| CrMediaPlayer_Player_Recorder | Player and Recorder |

## CrDeviceProperty_MediaSLOT2Player

Get the Media SLOT2 Player

| Parameter Code | Explanation |
|---|---|
| CrMediaPlayer_None | None |
| CrMediaPlayer_Player | Player |
| CrMediaPlayer_Recorder | Recorder |
| CrMediaPlayer_Player_Recorder | Player and Recorder |

## CrDeviceProperty_BatteryRemainDisplayUnit

Get/Set the Battery Remain Display Unit

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrBatteryRemainDisplayUnit_minute | minute |
| CrBatteryRemainDisplayUnit_percent | percent |
| CrBatteryRemainDisplayUnit_voltage | voltage |

## CrDeviceProperty_BatteryRemainingInMinutes

Get the Battery Remaining in minutes

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Unit is minute.<br><br>CrBatteryRemainingInMinutes_Untaken(0xFFFFFFFF) is untaken. |
| Variable | Max | |
| Variable | Step | |

## CrDeviceProperty_BatteryRemainingInVoltage

Get the Battery Remaining in voltage

| Parameter Code | Explanation | |
|---|---|---|
| Variable | Min | 1000 times the real value of Battery Remaining in voltage.<br><br>CrBatteryRemainingInVoltage_Untaken(0xFFFFFFFF) is untaken. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PowerSource

Get/Set the Power Source

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrPowerSource_DC | DC |
| CrPowerSource_Battery | Battery |

## CrDeviceProperty_DCVoltage

Get the DC voltage

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | 1000 times the real value of DC voltage.<br>CrDCVoltage_Untaken(0xFFFFFFFF) is untaken. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_FocusTouchSpotStatus

Get the Focus TouchSpot Status

| Parameter Code | Explanation |
|---|---|
| CrFocusTouchSpotStatus_Stopped | Stopped |
| CrFocusTouchSpotStatus_Running | Running |

## CrDeviceProperty_FocusTrackingStatus

Get the Focus Tracking Status

| Parameter Code | Explanation |
|---|---|
| CrFocusTrackingStatus_OFF | OFF |
| CrFocusTrackingStatus_Focusing | Focusing |
| CrFocusTrackingStatus_Tracking | Tracking |

## CrDeviceProperty_RecorderClipName

Get Recorder Clip Name Create by The Next Rec.

| Parameter Code | Explanation |
|---|---|
| String | Clip Name |

## CrDeviceProperty_RecorderControlMainSetting

Get the Recorder Control Main Setting

| Parameter Code | Explanation |
|---|---|
| CrRecorderControlSetting_RecDisable | Rec Disable |
| CrRecorderControlSetting_RecEnable | Rec Enable |

## CrDeviceProperty_RecorderControlProxySetting

Get/Set the Recorder Control Proxy Setting

For ILME-FX6, only gets are supported.

| Parameter Code | Explanation |
|---|---|
| CrRecorderControlSetting_RecDisable | Rec Disable |
| CrRecorderControlSetting_RecEnable | Rec Enable |

## CrDeviceProperty_RecorderStartMain

Get the Recorder Start Main

| Parameter Code | Explanation |
|---|---|
| CrRecorderStart_RecStartDisable | Rec Start Disable |
| CrRecorderStart_RecStartEnable | Rec Start Enable |

## CrDeviceProperty_RecorderStartProxy

Get the Recorder Start Proxy

| Parameter Code | Explanation |
|---|---|
| CrRecorderStart_RecStartDisable | Rec Start Disable |
| CrRecorderStart_RecStartEnable | Rec Start Enable |

## CrDeviceProperty_RecorderMainStatus

Get the Recorder Main Status

| Parameter Code | Explanation |
|---|---|
| CrRecorderStatus_Idle | Idle |
| CrRecorderStatus_Ready | Ready |
| CrRecorderStatus_PreparingToRecord | PreparingToRecord |
| CrRecorderStatus_Standby | Standby |
| CrRecorderStatus_Recording | Recording |
| CrRecorderStatus_Stopping | Stopping |

## CrDeviceProperty_RecorderProxyStatus

Get the Recorder Proxy Status

| Parameter Code | Explanation |
|---|---|
| CrRecorderStatus_Idle | Idle |
| CrRecorderStatus_Ready | Ready |
| CrRecorderStatus_PreparingToRecord | PreparingToRecord |
| CrRecorderStatus_Standby | Standby |
| CrRecorderStatus_Recording | Recording |
| CrRecorderStatus_Stopping | Stopping |

## CrDeviceProperty_RecorderExtRawStatus

Get the Recorder Ext Raw Status

| Parameter Code | Explanation |
|---|---|
| CrRecorderStatus_Idle | Idle |
| CrRecorderStatus_Ready | Ready |
| CrRecorderStatus_PreparingToRecord | PreparingToRecord |
| CrRecorderStatus_Standby | Standby |
| CrRecorderStatus_Recording | Recording |
| CrRecorderStatus_Stopping | Stopping |

## CrDeviceProperty_RecorderSaveDestination

Get the information of Recorder Save Destination

| Parameter Code | Explanation |
|---|---|
| CrRecorderSaveDestination_External | External |
| CrRecorderSaveDestination_Internal | Internal |
| CrRecorderSaveDestination_ExternalAndInternal | External & Internal |

## CrDeviceProperty_AssignableButtonIndicator1

Get the Assignable Button Indicator 1

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator2

Get the Assignable Button Indicator 2

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator3

Get the Assignable Button Indicator 3

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator4

Get the Assignable Button Indicator 4

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator5

Get the Assignable Button Indicator 5

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator6

Get the Assignable Button Indicator 6

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator7

Get the Assignable Button Indicator 7

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator8

Get the Assignable Button Indicator 8

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_AssignableButtonIndicator9

Get the Assignable Button Indicator 9

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_LensAssignableButtonIndicator1

Get the LensAssignable Button Indicator 1

| Parameter Code | Explanation |
|---|---|
| CrAssignableButtonIndicator_Off | OFF |
| CrAssignableButtonIndicator_On | ON |

## CrDeviceProperty_SoftwareVersion

Software Version.

| Parameter Code | Explanation |
|---|---|
| String | Software Version |

## CrDeviceProperty_MovieRecButtonToggleEnableStatus

Get the Movie Rec Button (Toggle) Enable Status

| Parameter Code | Explanation |
|---|---|
| CrMovieRecButtonToggle_Disable | Disable |
| CrMovieRecButtonToggle_Enable | Enable |

## CrDeviceProperty_RemoteTouchOperationEnableStatus

Get the Remote Touch Operation Enable Status

| Parameter Code | Explanation |
|---|---|
| CrRemoteTouchOperation_Disable | Disable |
| CrRemoteTouchOperation_Enable | Enable |

## CrDeviceProperty_CancelRemoteTouchOperationEnableStatus

Get the Cancel Remote Touch Operation Enable Status

| Parameter Code | Explanation |
|---|---|
| CrCancelRemoteTouchOperation_Disable | Disable |
| CrCancelRemoteTouchOperation_Enable | Enable |

## CrDeviceProperty_LensInformationEnableStatus

Get the Lens Information Enable Status

| Parameter Code | Explanation |
|---|---|
| CrLensInformation_Disable | Disable |
| CrLensInformation_Enable | Enable |

## CrDeviceProperty_FollowFocusPositionSetting

Get/Set the Follow Focus Position

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | The Focus Position can be changed within this range. This CurrentValue will be the requested value. Check the actual Focus Position with CrDeviceProperty_FollowFocusPositionCurrentValue. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_FollowFocusPositionCurrentValue

Get the Follow Focus Position Current Value

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | CurrentValue (normalized value) of the Focus Position.<br><br>Note:<br>Can be converted from normalized values to Focus Position (meters/feet) using data taken with GetLensInformation().<br>Focus drive suitable for movie recording. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_FocusBracketShotNumber

Get/Set the Focus Bracket Shot Num

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Focus Bracket The number of shots to take. |
| Variable | max | |
| 1 | step | |

## CrDeviceProperty_FocusBracketFocusRange

Get/Set the Focus Bracket Focus Range

| Parameter Code | Explanation | |
|---|---|---|
| Variable | min | Focus Bracket Focus range when shooting.<br>Variable follows model specifications. |
| Variable | max | |
| 1 | step | |

## CrDeviceProperty_FocusBracketShootingStatus

Get the Focus Bracket Shooting Status

| Parameter Code | Explanation |
|---|---|
| CrFocusBracket_NotShooting | Not Shooting |
| CrFocusBracket_Shooting | Shooting |

## CrDeviceProperty_FunctionOfTouchOperation

Get/Set the Function of Touch Operation

| Parameter Code | Explanation | |
|---|---|---|
| CrFunctionOfTouchOperation_Off | OFF | |
| CrFunctionOfTouchOperation_Shutter | Only for models do not support touch AE | |
| | | Touch Shutter |
| CrFunctionOfTouchOperation_Focus | | Touch Focus |
| CrFunctionOfTouchOperation_Tracking | | Touch Tracking |
| CrFunctionOfTouchOperation_AE | Only for touch AE support models | |
| | | Touch AE |
| CrFunctionOfTouchOperation_ShutterAndAEOn | | Touch Shutter and Touch AE ON |
| CrFunctionOfTouchOperation_ShutterAndAEOff | | Touch Shutter and Touch AE OFF |
| CrFunctionOfTouchOperation_FocusAndAEOn | | Touch Focus and Touch AE ON |
| CrFunctionOfTouchOperation_FocusAndAEOff | | Touch Focus and Touch AE OFF |
| CrFunctionOfTouchOperation_TrackingAndAEOn | | Touch Tracking and Touch AE ON |
| CrFunctionOfTouchOperation_TrackingAndAEOff | | Touch Tracking and Touch AE OFF |

## CrDeviceProperty_Movie_ProxyFileFormat

Get/Set the Proxy File Format(Movie)

| Parameter Code | Explanation |
|---|---|
| CrFileFormatMovie_XAVC_S_HD | XAVC S HD |
| CrFileFormatMovie_XAVC_HS_HD | XAVC HS HD |
| CrFileFormatMovie_XAVC_S_I_DCI_4K | XAVC S-I DCI 4K |

## CrDeviceProperty_ExtendedInterfaceMode

Get/Set the Extended Interface Mode

Extended interface for Sony's Camera Remote SDK supports shutter trigger by electrical signal and remote power On/Off*.

| Parameter Code | Explanation |
|---|---|
| CrExtendedInterfaceMode_Off | OFF |
| CrExtendedInterfaceMode_On | ON |

*: Extended interface for Sony's Camera Remote SDK is compatible only with the camera that has a USB Type-C® connector. To use this function, a separate device must be prepared and connected. For details, please download the following link.
https://support.d-imaging.sony.co.jp/app/sdk/extended_interface/en/index.html

## CrDeviceProperty_SQFrameRate

Get/Set the S&Q Frame Rate

| Parameter Code | Explanation |
|---|---|
| CrSQFrameRate_Invalid | Invalid |
| Other than above values | The range of frame rate is 1fps ~ 60fps, and 100fps / 120fps / 150fps / 180fps / 200fps / 240fps. |

## CrDeviceProperty_SQRecordingFrameRateSetting

Get/Set the S&Q Recording Frame Rate Setting

| Parameter Code | Explanation |
|---|---|
| CrRecordingFrameRateSettingMovie_120p | 120p<br>Actual frequency might be 119.88. |
| CrRecordingFrameRateSettingMovie_100p | 100p |
| CrRecordingFrameRateSettingMovie_60p | 60p<br>Actual frequency might be 59.94. |
| CrRecordingFrameRateSettingMovie_50p | 50p |
| CrRecordingFrameRateSettingMovie_30p | 30p<br>Actual frequency might be 29.97. |
| CrRecordingFrameRateSettingMovie_25p | 25p |
| CrRecordingFrameRateSettingMovie_24p | 24p<br>Actual frequency might be 23.98. |
| CrRecordingFrameRateSettingMovie_23_98p | 23.98p |
| CrRecordingFrameRateSettingMovie_29_97p | 29.97p |
| CrRecordingFrameRateSettingMovie_59_94p | 59.94p |
| CrRecordingFrameRateSettingMovie_24_00p | 24.00p |
| CrRecordingFrameRateSettingMovie_119_88p | 119.88p |

See Table fr-1/2 in CrDeviceProperty_Movie_Recording_FrameRateSetting

## CrDeviceProperty_SQRecordingSetting

Get/Set the S&Q Recording Setting

| Parameter Code | Explanation |
|---|---|
| CrRecordingSettingMovie_Invalid | Invalid |
| CrRecordingSettingMovie_600M_422_10bit | 600M 422 10bit |
| CrRecordingSettingMovie_500M_422_10bit | 500M 422 10bit |
| CrRecordingSettingMovie_400M_420_10bit | 400M 420 10bit |
| CrRecordingSettingMovie_300M_422_10bit | 300M 422 10bit |
| CrRecordingSettingMovie_280M_422_10bit | 280M 422 10bit |
| CrRecordingSettingMovie_250M_422_10bit | 250M 422 10bit |
| CrRecordingSettingMovie_240M_422_10bit | 240M 422 10bit |
| CrRecordingSettingMovie_222M_422_10bit | 222M 422 10bit |
| CrRecordingSettingMovie_200M_422_10bit | 200M 422 10bit |
| CrRecordingSettingMovie_200M_420_10bit | 200M 420 10bit |
| CrRecordingSettingMovie_200M_420_8bit | 200M 420 8bit |
| CrRecordingSettingMovie_185M_422_10bit | 185M 422 10bit |
| CrRecordingSettingMovie_150M_420_10bit | 150M 420 10bit |
| CrRecordingSettingMovie_150M_420_8bit | 150M 420 8bit |
| CrRecordingSettingMovie_140M_422_10bit | 140M 422 10bit |
| CrRecordingSettingMovie_111M_422_10bit | 111M 422 10bit |
| CrRecordingSettingMovie_100M_422_10bit | 100M 422 10bit |
| CrRecordingSettingMovie_100M_420_10bit | 100M 420 10bit |
| CrRecordingSettingMovie_100M_420_8bit | 100M 420 8bit |
| CrRecordingSettingMovie_93M_422_10bit | 93M 422 10bit |
| CrRecordingSettingMovie_89M_422_10bit | 89M 422 10bit |
| CrRecordingSettingMovie_75M_420_10bit | 75M 420 10bit |
| CrRecordingSettingMovie_60M_420_8bit | 60M 420 8bit |
| CrRecordingSettingMovie_50M_422_10bit | 50M 422 10bit |
| CrRecordingSettingMovie_50M_420_10bit | 50M 420 10bit |
| CrRecordingSettingMovie_50M_420_8bit | 50M 420 8bit |
| CrRecordingSettingMovie_45M_420_10bit | 45M 420 10bit |
| CrRecordingSettingMovie_30M_420_10bit | 30M 420 10bit |
| CrRecordingSettingMovie_25M_420_8bit | 25M 420 8bit |
| CrRecordingSettingMovie_16M_420_8bit | 16M 420 8bit |
| CrRecordingSettingMovie_520M_422_10bit | 520M 422 10bit |
| CrRecordingSettingMovie_260M_422_10bit | 260M 422 10bit |

## CrDeviceProperty_AudioRecording

Get/Set the Audio Recording

| Parameter Code | Explanation |
|---|---|
| CrAudioRecording_Off | Off |
| CrAudioRecording_On | On |

## CrDeviceProperty_AudioInputMasterLevel

Get/Set the Audio Input Master Level

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | You can adjust the audio recording level. |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_TimeCodePreset

Get/Set the Time Code Preset

| Parameter Code | | Explanation |
|---|---|---|
| 0x00000000 | min | Bit24-Bit31 : hour<br>Bit16-Bit23 : minute<br>Bit8-Bit15 : second<br>Bit0-Bit7 : frame<br><br>The time code can be set between the following range.<br>- When [60p] is selected: 00:00:00.00 to 23:59:59.29<br>- When [24p] is selected, you can select the last two digits of the time code in multiples of four from 00 to 23 frames.<br>- When [50p] is selected: 00:00:00.00 to 23:59:59.24 |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_TimeCodeFormat

Get/Set the Time Code Format

| Parameter Code | Explanation |
|---|---|
| CrTimeCodeFormat_DF | DF |
| CrTimeCodeFormat_NDF | NDF |

## CrDeviceProperty_TimeCodeRun

Get/Set the Time Code Run

| Parameter Code | Explanation |
|---|---|
| CrTimeCodeRun_RecRun | Rec Run |
| CrTimeCodeRun_FreeRun | Free Run |

## CrDeviceProperty_TimeCodeMake

Get/Set the Time Code Make

| Parameter Code | Explanation |
|---|---|
| CrTimeCodeMake_Preset | Preset |
| CrTimeCodeMake_Regenerate | Regenerate |

## CrDeviceProperty_UserBitPreset

Get/Set the User Bit Preset

| Parameter Code | | Explanation |
|---|---|---|
| 0x00000000 | min | Saved to file in LSB First order. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_UserBitTimeRec

Get/Set the User Bit Time Rec

| Parameter Code | Explanation |
|---|---|
| CrUserBitTimeRec_Off | Off |
| CrUserBitTimeRec_On | On |

## CrDeviceProperty_ImageStabilizationSteadyShot

Get/Set the Image Stabilization Steady Shot

| Parameter Code | Explanation |
|---|---|
| CrImageStabilizationSteadyShot_Off | Off |
| CrImageStabilizationSteadyShot_On | On |

## CrDeviceProperty_Movie_ImageStabilizationSteadyShot

Get/Set the Image Stabilization Steady Shot(Movie)

| Parameter Code | Explanation |
|---|---|
| CrImageStabilizationSteadyShotMovie_Off | Off |
| CrImageStabilizationSteadyShotMovie_Standard | Standard |
| CrImageStabilizationSteadyShotMovie_Active | Active |
| CrImageStabilizationSteadyShotMovie_DynamicActive | Dynamic active |

## CrDeviceProperty_SilentMode

Get/Set the Silent Mode

| Parameter Code | Explanation |
|---|---|
| CrSilentMode_Off | Off |
| CrSilentMode_On | On |

## CrDeviceProperty_SilentModeApertureDriveInAF

Get/Set the Silent Mode Aperture Drive in AF

| Parameter Code | Explanation |
|---|---|
| CrSilentModeApertureDriveInAF_NotTarget | Not Target |
| CrSilentModeApertureDriveInAF_Standard | Standard |
| CrSilentModeApertureDriveInAF_SilentPriority | Silent Priority |

## CrDeviceProperty_SilentModeShutterWhenPowerOff

Get/Set the Silent Mode Shutter When Power Off

| Parameter Code | Explanation |
|---|---|
| CrSilentModeShutterWhenPowerOff_NotTarget | Not Target |
| CrSilentModeShutterWhenPowerOff_Off | Off |

## CrDeviceProperty_SilentModeAutoPixelMapping

Get/Set the Silent Mode Auto Pixel Mapping

| Parameter Code | Explanation |
|---|---|
| CrSilentModeAutoPixelMapping_NotTarget | Not Target |
| CrSilentModeAutoPixelMapping_Off | Off |

## CrDeviceProperty_ShutterType

Get/Set the Shutter Type

| Parameter Code | Explanation |
|---|---|
| CrShutterType_Auto | Auto |
| CrShutterType_MechanicalShutter | Mechanical Shutter |
| CrShutterType_ElectronicShutter | Electronic Shutter |

## CrDeviceProperty_PictureProfile

Get/Set the Picture Profile

| Parameter Code | Explanation |
|---|---|
| CrPictureProfile_Off | Picture Profile Off |
| CrPictureProfile_Number1 | Picture Profile number 1 |
| CrPictureProfile_Number2 | Picture Profile number 2 |
| CrPictureProfile_Number3 | Picture Profile number 3 |
| CrPictureProfile_Number4 | Picture Profile number 4 |
| CrPictureProfile_Number5 | Picture Profile number 5 |
| CrPictureProfile_Number6 | Picture Profile number 6 |
| CrPictureProfile_Number7 | Picture Profile number 7 |
| CrPictureProfile_Number8 | Picture Profile number 8 |
| CrPictureProfile_Number9 | Picture Profile number 9 |
| CrPictureProfile_Number10 | Picture Profile number 10 |
| CrPictureProfile_Number11 | Picture Profile number 11 |
| CrPictureProfile_LUT_Number1 | Picture Profile LUT number 1 |
| CrPictureProfile_LUT_Number2 | Picture Profile LUT number 2 |
| CrPictureProfile_LUT_Number3 | Picture Profile LUT number 3 |
| CrPictureProfile_LUT_Number4 | Picture Profile LUT number 4 |

## CrDeviceProperty_PictureProfile_BlackLevel

Get/Set the Picture Profile Black Level

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Black Level for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_Gamma

Get/Set the Picture Profile Gamma

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileGamma_Movie | Movie |
| CrPictureProfileGamma_Still | Still |
| CrPictureProfileGamma_S_Cinetone | S-Cinetone |
| CrPictureProfileGamma_Cine1 | Cine1 |
| CrPictureProfileGamma_Cine2 | Cine2 |
| CrPictureProfileGamma_Cine3 | Cine3 |
| CrPictureProfileGamma_Cine4 | Cine4 |
| CrPictureProfileGamma_ITU709 | ITU709 |
| CrPictureProfileGamma_ITU709_800 | ITU709(800%) |

| | |
|---|---|
| CrPictureProfileGamma_S_Log2 | S-Log2 |
| CrPictureProfileGamma_S_Log3 | S-Log3 |
| CrPictureProfileGamma_HLG | HLG |
| CrPictureProfileGamma_HLG1 | HLG1 |
| CrPictureProfileGamma_HLG2 | HLG2 |
| CrPictureProfileGamma_HLG3 | HLG3 |

## CrDeviceProperty_PictureProfile_BlackGammaRange

Get/Set the Picture Profile Black Gamma Range

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileBlackGammaRange_Wide | Wide |
| CrPictureProfileBlackGammaRange_Middle | Middle |
| CrPictureProfileBlackGammaRange_Narrow | Narrow |

## CrDeviceProperty_PictureProfile_BlackGammaLevel

Get/Set the Picture Profile Black Gamma Level

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Black Gamma Level for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_KneeMode

Get/Set the Picture Profile Knee Mode

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileKneeMode_Auto | Auto |
| CrPictureProfileKneeMode_Manual | Manual |

## CrDeviceProperty_PictureProfile_KneeAutoSet_MaxPoint

Get/Set the Picture Profile Knee AutoSet MaxPoint

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileKneeSetPoint_Invalid | Invalid |
| Other than above values | Knee AutoSet MaxPoint for Picture Profile<br><br>100 times the value of MaxPoint(%)<br>ex) 0x2616 = 97.50% |

## CrDeviceProperty_PictureProfile_KneeAutoSet_Sensitivity

Get/Set the Picture Profile Knee AutoSet Sensitivity

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileKneeAutoSetSensitivity_Low | Low |
| CrPictureProfileKneeAutoSetSensitivity_Mid | Middle |
| CrPictureProfileKneeAutoSetSensitivity_High | High |

## CrDeviceProperty_PictureProfile_KneeManualSet_Point

Get/Set the Picture Profile Knee ManualSet Point

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileKneeSetPoint_Invalid | Invalid |
| Other than above values | Knee ManualSet Point for Picture Profile<br><br>100 times the value of Point(%)<br>ex) 0x2616 = 97.50% |

## CrDeviceProperty_PictureProfile_KneeManualSet_Slope

Get/Set the Picture Profile Knee ManualSet Slope

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Knee ManualSet Slope for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorMode

Get/Set the Picture Profile Color Mode

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileColorMode_Movie | Movie |
| CrPictureProfileColorMode_Still | Still |
| CrPictureProfileColorMode_S_Cinetone | S-Cinetone |
| CrPictureProfileColorMode_Cinema | Cinema |
| CrPictureProfileColorMode_Pro | Pro |
| CrPictureProfileColorMode_ITU709_Matrix | ITU709 Matrix |
| CrPictureProfileColorMode_BlackWhite | Black&White |
| CrPictureProfileColorMode_S_Gamut3_Cine | S-Gamut3.Cine |
| CrPictureProfileColorMode_S_Gamut3 | S-Gamut3 |
| CrPictureProfileColorMode_BT_2020 | BT.2020 |
| CrPictureProfileColorMode_709 | 709 |
| CrPictureProfileColorMode_S_Gamut | S-Gamut |

## CrDeviceProperty_PictureProfile_Saturation

Get/Set the Picture Profile Saturation

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Saturation for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorPhase

Get/Set the Picture Profile Color Phase

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Phase for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthRed

Get/Set the Picture Profile Color Depth Red

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Red for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthGreen

Get/Set the Picture Profile Color Depth Green

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Green for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthBlue

Get/Set the Picture Profile Color Depth Blue

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Blue for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthCyan

Get/Set the Picture Profile Color Depth Cyan

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Cyan for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthMagenta

Get/Set the Picture Profile Color Depth Magenta

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Magenta for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_ColorDepthYellow

Get/Set the Picture Profile Color Depth Yellow

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Color Depth Yellow for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_DetailLevel

Get/Set the Picture Profile Detail Level

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Detail Level for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_DetailAdjustMode

Get/Set the Picture Profile Detail Adjust Mode

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileDetailAdjustMode_Auto | Auto |
| CrPictureProfileDetailAdjustMode_Manual | Manual |

## CrDeviceProperty_PictureProfile_DetailAdjustVHBalance

Get/Set the Picture Profile Detail Adjust V/H Balance

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Detail Adjust V/H Balance for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_DetailAdjustBWBalance

Get/Set the Picture Profile Detail Adjust B/W Balance

| Parameter Code | Explanation |
|---|---|
| Variable | Detail Adjust B/W Balance for Picture Profile |

## CrDeviceProperty_PictureProfile_DetailAdjustLimit

Get/Set the Picture Profile Detail Adjust Limit

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Detail Adjust Limit for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_DetailAdjustCrispening

Get/Set the Picture Profile Detail Adjust Crispening

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Detail Adjust Crispening for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_DetailAdjustHiLightDetail

Get/Set the Picture Profile Detail Adjust Hi-Light Detail

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Detail Adjust Hi-Light Detail for Picture Profile |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_PictureProfile_Copy

Get/Set the Copy Picture Profile

| Parameter Code | Explanation |
|---|---|
| Variable | Select the copy destination Picture Profile number for Picture Profile selected in "Picture Profile" |

## CrDeviceProperty_PictureProfileResetEnableStatus

Get the Picture Profile Reset Enable Status

| Parameter Code | Explanation |
|---|---|
| CrPictureProfileReset_Disable | Disable |
| CrPictureProfileReset_Enable | Enable |

## CrDeviceProperty_CreativeLook

Get/Set the Creative Look

| Parameter Code | Explanation |
|---|---|
| CrCreativeLook_ST | ST |
| CrCreativeLook_PT | PT |
| CrCreativeLook_NT | NT |
| CrCreativeLook_VV | VV |
| CrCreativeLook_VV2 | VV2 |
| CrCreativeLook_FL | FL |
| CrCreativeLook_IN | IN |
| CrCreativeLook_SH | SH |
| CrCreativeLook_BW | BW |
| CrCreativeLook_SE | SE |
| CrCreativeLook_CustomLookOffset | CustomLook offset.<br>When the upper 8 bits are 0x01, it means CustomLook.<br>e.g.) 0x0101 = CustomLook 1<br>e.g.) 0x0103 = CustomLook 3 |

Refer to the camera's help guide for details on each setting

## CrDeviceProperty_CreativeLook_Contrast

Get/Set the Creative Look Contrast

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Contrast for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Highlights

Get/Set the Creative Look Highlights

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Highlights for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Shadows

Get/Set the Creative Look Shadows

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Shadows for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Fade

Get/Set the Creative Look Fade

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Fade for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Saturation

Get/Set the Creative Look Saturation

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Saturation for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Sharpness

Get/Set the Creative Look Sharpness

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Sharpness for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_SharpnessRange

Get/Set the Creative Look Sharpness Range

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Sharpness Range for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_Clarity

Get/Set the Creative Look Clarity

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | Clarity for Creative Look |
| Variable | max | |
| Variable | step | |

## CrDeviceProperty_CreativeLook_CustomLook

Get/Set the Custom Look in Creative Look

| Parameter Code | Explanation |
|---|---|
| CrCreativeLook_ST | ST |
| CrCreativeLook_PT | PT |
| CrCreativeLook_NT | NT |
| CrCreativeLook_VV | VV |
| CrCreativeLook_VV2 | VV2 |
| CrCreativeLook_FL | FL |
| CrCreativeLook_IN | IN |
| CrCreativeLook_SH | SH |
| CrCreativeLook_BW | BW |
| CrCreativeLook_SE | SE |

## CrDeviceProperty_CreativeLookResetEnableStatus

Get the Creative Look Reset Enable Status

| Parameter Code | Explanation |
|---|---|
| CrCreativeLookReset_Disable | Disable |
| CrCreativeLookReset_Enable | Enable |

## CrDeviceProperty_ProxyRecordingSetting

Get/Set the Proxy Record Setting

| Parameter Code | Explanation |
|---|---|
| CrProxyRecordingSetting_Invalid | Invalid |
| CrProxyRecordingSetting_16M_420_10bit | 16M 420 10bit |
| CrProxyRecordingSetting_9M_420_10bit | 9M 420 10bit |
| CrProxyRecordingSetting_6M_420_8bit | 6M 420 8bit |

## CrDeviceProperty_Movie_IntervalRec_CountDownIntervalTime

Get the Interval REC(Movie) Count Down Interval Time

| Parameter Code | | Explanation |
|---|---|---|
| 0x00000000 | min | The real value of Interval REC(Time Lapse Movie) count down interval time. Unit is second<br><br>e.g.) 0x00000001 = 1 sec<br>e.g.) 0x0000003C = 1 min |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_Movie_IntervalRec_RecordingDuration

Get the Interval REC(Movie) Recording Duration

| Parameter Code | | Explanation |
|---|---|---|
| 0x00000000 | min | Interval REC(Time Lapse Movie) recorded clip length. Unit is second.<br><br>For example, if you start shooting at 30 fps at 1 second intervals, CurrentValue will increase by 1 for every 30 seconds elapsed. |
| 0xFFFFFFFF | max | |
| 0x00000001 | step | |

## CrDeviceProperty_PixelMappingEnableStatus

Get the Pixel Mapping Enable Status

| Parameter Code | Explanation |
|---|---|
| CrPixelMapping_Disable | Disable |
| CrPixelMapping_Enable | Enable |

## CrDeviceProperty_TimeCodePresetResetEnableStatus

Get the Time Code Preset Reset Enable Status

| Parameter Code | Explanation |
|---|---|
| CrTimeCodePresetReset_Disable | Disable |
| CrTimeCodePresetReset_Enable | Enable |

## CrDeviceProperty_UserBitPresetResetEnableStatus

Get the User Bit Preset Reset Enable Status

| Parameter Code | Explanation |
|---|---|
| CrUserBitPresetReset_Disable | Disable |
| CrUserBitPresetReset_Enable | Enable |

## CrDeviceProperty_SensorCleaningEnableStatus

Get the Sensor Cleaning Enable Status

| Parameter Code | Explanation |
|---|---|
| CrSensorCleaning_Disable | Disable |
| CrSensorCleaning_Enable | Enable |

## CrDeviceProperty_LensVersionNumber

Get the Lens Version Number

| Parameter Code | Explanation |
|---|---|
| String | Lens version major number.<br>For example, when the major number of the lens version is 1, "01" is set.<br><br>CrLensVersionNumber_Untaken(blank) if the lens version cannot be obtained. |

## CrDeviceProperty_DeviceOverheatingState

Get the Device Overheating State

| Parameter Code | | Explanation |
|---|---|---|
| Variable | min | The current value can be one of the following three.<br><br>CrDeviceOverheatingState_NotOverheating<br>CrDeviceOverheatingState_PreOverheating<br>CrDeviceOverheatingState_Overheating |
| Variable | max | |
| Variable | step | |

# Tips / Trouble Shooting

## Shutter Release

If you struggle to make “Shutter Release” success in a remote control, please try to set camera settings “Exposure Program Mode” with “M(Manual)” and “FocusMode” with “MF(Manual Focus)”.
∵As camera accepts “Shutter release control” after coming into focus in several Auto Focus modes, sometimes focus mode setting, focus area setting, and shooting environmental conditions prevent camera to accept “Shutter Release”.

### Remote Control Settings Example

1. “CrDeviceProperty_PriorityKeySettings” with “CrPriorityKey_PCRemote”
2. “CrDeviceProperty_ExposureProgramMode” with “CrExposure_M_Manual”
3. “CrDeviceProperty_FocusMode” with “CrFocus_MF”
4. “CrCommandId_Release” with “CrCommandParam_Down”
5. “CrCommandId_Release” with “CrCommandParam_Up”

Also, memory card full situation prevents shutter release from execution, so it is recommended to prepare enough space in the memory card and / or prepare dual memory cards before remote control.

## Shutter Half Release / Auto Focus

If you struggle to make “Shutter Half Release” success and come into focus successfully in remote controls, please try to set camera settings “FocusMode” with “AF-S”, and “FocusArea” with “Wide”.
∵As camera occasionally takes time relatively to come into focus depends on settings and shooting environmental conditions in several auto focus modes, above settings have relatively wide acceptance to come into focus.

### Remote Control Settings Example

1. “CrDeviceProperty_PriorityKeySettings” with “CrPriorityKey_PCRemote”
2. “CrDeviceProperty_FocusMode” with “CrFocus_AF_S”
3. “CrDeviceProperty_FocusArea” with “CrFocusArea_Wide”
4. “CrDeviceProperty_S1” with “CrLockIndicator_Locked”
5. “CrDeviceProperty_S1” with “CrLockIndicator_Unlocked”

## Manual Focus

If you struggle to control focus manually in remote controls, please try to set camera settings “FocusMode” with “MF(Manual Focus)”.

### Remote Control Settings Example

1. “CrDeviceProperty_PriorityKeySettings” with “CrPriorityKey_PCRemote”
2. “CrDeviceProperty_FocusMode” with “CrFocus_MF”

# Device Property

If you struggle to change camera settings, it is recommended to check enable flag in each DeviceProperty by sending GetDeviceProperties and receiving the latest information before sending SetDeviceProperty. ∵As the specification of camera products, camera settings have exclusive conditions. For example, focus control Near/Far is not acceptable in Focus Mode “AF-S”. In order to identify whether an issue is coming from remote control related or camera settings acceptable/unacceptable conditions, you better try what you want to do first w/o remote control but w/ direct camera operation by camera buttons / menu settings. Then copy operations with remote control. “Help Guide” for each product may help you to understand the specification of camera products including acceptable/unacceptable conditions of settings.

### Remote Control Settings Example

1. “GetDeviceProperties” with “CrDevicePropertyCode”
2. Check “CrPropertyEnableFlag” of “CrDeviceProperty”
3. “SetDeviceProperty” with “CrDevicePropertyCode”

Also, it is recommended to set a value from candidate values list in each DeviceProperty after sending GetDeviceProperties and receiving the latest information before sending SetDeviceProperty. ∵As the specification of camera products, camera settings have variable acceptance for value depends on settings and shooting environmental conditions. For example, acceptable F number value varies depends on the lens attached to the camera, other settings, and the shooting environmental conditions.

### Remote Control Settings Example

1. “GetDeviceProperties” with “CrDevicePropertyCode”
2. Check “valuesSize” and “values” of “CrDeviceProperty”
3. “SetDeviceProperty” with “CrDevicePropertyCode”

Some of DeviceProperties are originally assigned on HardKeys of the camera product, and in these cases, need to change KeyPriority from “CameraPosition” to “PCRemote” before sending SetDeviceProperty. This applies to “ExposureProgramMode”, “FocusMode” and “Still Capture Mode(Drive Mode)”.

### Remote Control Settings Example

1. “CrDeviceProperty_PriorityKeySettings” with “CrPriorityKey_PCRemote”
2. “SetDeviceProperty” with “CrDevicePropertyCode”

# Transfer of shot images preparation

If you struggle to transfer shot images to PC, please check if you changed “StillImageStoreDestination” before shutter button release. You can select from HostPC/MemoryCard/HostPCAndMemoryCard. When you transfer shot images to PC, you need to change it to HostPC/HostPCAndMemoryCard beforehand.

### Remote Control Settings Example

1. “CrDeviceProperty_StillImageStoreDestination” with “CrStillImageStoreDestination_HostPCAndMemoryCard(or _HostPC)”
2. “CrCommandId_Release” with “CrCommandParam_Down”
3. “CrCommandId_Release” with “CrCommandParam_Up”
4. Check the folder set by SetSaveInfo() and open image files transferred to PC.

Please note that if once Host PC transfer is set like above, camera side also starts preparing and sending out image files, it is recommended to disconnect after finishing transfer of all images shot on

the camera. If disconnected before transfer finishes, camera and PC restart to transfer after reconnection, except for camera power off or physical disconnection case.

## Selected Media Format

If Still Image Save Destination is Host Device, recording media cannot be initialized.

If you want to initialize it, change Still Image Save Destination to Camera or Host Device and Camera.

### Remote Control Settings Example

1. “CrDeviceProperty_StillImageStoreDestination” with “CrStillImageStoreDestination_HostPCAndMemoryCard(or _MemoryCard)”

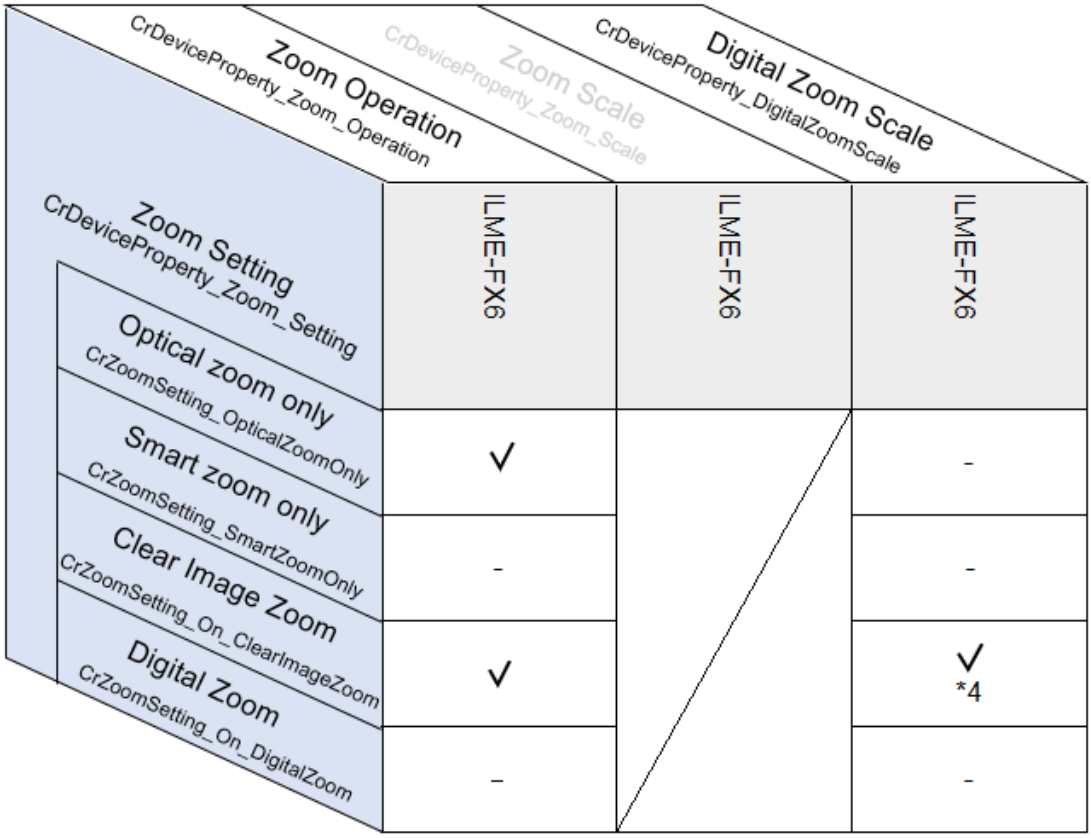
## Zoom Operation / Zoom Scale

Shows the relationship the Zoom Operation property, the Zoom Scale property, and the Digital Zoom Scale property, and the Zoom Setting property.

Table z-1. All models except ILME-FX6

| Zoom Setting (CrDeviceProperty_Zoom_Setting) | Zoom Operation (CrDeviceProperty_Zoom_Operation) | | Zoom Scale (CrDeviceProperty_Zoom_Scale) | | Digital Zoom Scale (CrDeviceProperty_DigitalZoomScale) | |
|---|---|---|---|---|---|---|
| | DSC-RX0M2 | Other Models | DSC-RX0M2 | Other Models | DSC-RX0M2 | Other Models |
| Optical zoom only (CrZoomSetting_OpticalZoomOnly) | - | ✓ *1 | - | ✓ *2*3 | | |
| Smart zoom only (CrZoomSetting_SmartZoomOnly) | ✓ *3 | - | ✓ *3 | - | | |
| Clear Image Zoom (CrZoomSetting_On_ClearImageZoom) | ✓ | ✓ *1 | ✓ | ✓ *2 | | |
| Digital Zoom (CrZoomSetting_On_DigitalZoom) | ✓ | ✓ *1 | ✓ | ✓ *2 | | |

Table z-2. For ILME-FX6




*1 : Power Zoom Lenses such as SELP1650, SELP18105G, SELP18110G, SELP18200 and SELP28135G.
*2 : When not using Power Zoom Lenses.
*3 : When the Image Size is “CrImageSize_M” or ”CrImageSize_S”.
*4 : Get only.

## Live View

If you struggle to have stable live view images, please check following factors affect to transmission of LiveView images.

-Traffic on the physical connection between PC and camera, such as HUB connection, not related devices connection, and so on.

-Traffic on the communicational connection between PC and camera, such as frequent shutter releases and transfers, frequent Get/Set device properties, and so on.

-Performance of PC (CPU power, memory resource, device specification, etc. ).

-Some functions to be disabled they can be processing loads to CPU on the Single Board Computer, such as Wi-Fi function.

If you prefer stable frame rate of live view images, minimizing image size of Live View images (and/or capturing images), reducing frequency of shutter release, stopping capturing images, and stopping transferring images to PC contributes to it.

# Camera Settings Saving

After changing camera settings, if you detach a battery from a camera (or stop power supply through power supply cable) without completing power off sequence with camera power button control, there is no guarantee that camera setting changes are saved. It is recommended to complete power off sequence with camera power button control at least once after you change camera settings, if you prefer to resume camera settings as you changed for next use.

# Focus Magnifier Setting

If you want to update “Focus Magnifier Setting”, implement the following steps.
refs. Device Properties and Live View Properties

1. Get a list of properties using the GetDeviceProperties
2. Look for "Focus Magnifier Setting" from the list of properties to find out the list of selectable focus magnification

*Example:*

```
switch (property->GetCode()) {
case CrDeviceProperty_Focus_Magnifier_Setting:
    CrInt64u currentvalue = static_cast<CrInt64u>(property->GetCurrentValue());
    CrInt32u ratioNow = (currentvalue >> 32);
    CrInt16u xNow = ((currentvalue >> 16) & 0xFFFF);
    CrInt16u yNow = (currentvalue & 0xFFFF);
    CrInt32u valCount = property->GetValueSize() / sizeof(CrInt64u);
    CrInt64u* ratioSetList = new CrInt64u[valCount];
    memcpy(ratioSetList, property->GetValues(),(size_t)property->GetValueSize());
```

3. Use the GetLiveViewProperties to get a list of Live View properties
4. Look for "CrMagPosInfo" in the retrieved list of Live View properties to find out the range of configurable positions

*Example:*

```
switch (lvproperty->GetCode()) {
case CrLiveViewProperty_Focus_Magnifier_Position:
    if (CrFrameInfoType::CrFrameInfoType_Magnifier_Position == lvproperty->GetFrameInfoType()) {
        CrMagPosInfo *pPosInfo = (CrMagPosInfo*)(lvproperty->GetValue());
        posXmax = pPosInfo->xDenominator;
        posYmax = pPosInfo->yDenominator;
```

5. Create a 64 bit value by combining the magnification rate obtained in step 2 and the coordinates that do not exceed the range obtained in step 4
6. Call SetDeviceProperty with the value you created in step 5

*Example:*

```
CrInt32u setX = 200; // Between 0 and (posXmax-1)
CrInt32u setY = 150; // Between 0 ant (posYmax-1)
CrInt64u setvalue = (ratioSetList[2] & 0xFFFFFFFF00000000) | (setX << 16) | setY;
CrDeviceProperty prop;
prop.SetCode(CrDeviceProperty_Focus_Magnifier_Setting);
prop.SetCurrentValue(setvalue);
prop.SetValueType(CrDataType_UInt64);
SetDeviceProperty(deviceHandle, &prop);
```

# About the Monitor DISP(Screen Display) for camera body

Shows the relationship the Monitor DISP Mode Candidate property, and the Monitor DISP Mode Setting property.

| CrDevicePropertyCode | | | | | | | | | | Explanation |
|---|---|---|---|---|---|---|---|---|---|---|
| CrDeviceProperty_DispModeCandidate | | | | | | | | | | |
| over 0x00000100 | 0x00000100 | 0x00000080 | 0x00000040 | 0x00000020 | 0x00000010 | 0x00000008 | 0x00000004 | 0x00000002 | 0x00000001 | CrDispModeBitNum |
| spare | Monitor Off | For viewfinder | No Disp. Info. Exposure:TimeOut | No Disp. Info. Exposure:On | No Disp. Info. | Level | Histogram | Display All Info. | Graphic Display | Display Only.<br><br>0x00000020 and 0x00000040 are exclusive.<br><br>Some items may not be displayed depending on the model. |
| CrDeviceProperty_DispModeSetting | | | | | | | | | | |
| - | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | The minimum value is 0x00000001 and the maximum value is 0x000001FF. |

# How to use LensInformation

Get a table for converting CrDeviceProperty_FollowFocusPositionSetting and CrDeviceProperty_FollowFocusPositionCurrentValue to Focus position (meters/feet).
Valid only when compatible lens is attached.

The following are available when CrDeviceProperty_LensInformationEnableStatus is set to Enable.

*Example:*

```
std::vector<SCRSDK::CrLensInformation*> m_lensInfo;

// Call the request
CrError err = SCRSDK::RequestLensInformation(handle);
```

*When the OnWarning callback notifies you of success:*

```
CrInt32u numOfList= 0;
SCRSDK::CrLensInformation* list = nullptr;

CrError err = SCRSDK::GetLensInformation(
                    handle,
                    &list,
                    &numOfList);

if (CR_SUCCEEDED(err) && 0 < numOfList) {

    for (int i = 0; i < numOfList; ++i) {
        auto item = new SCRSDK::CrLensInformation();
        item->normalizedValue = list[i].normalizedValue;
        item->focusPosition = list[i].focusPosition;
        m_lensInfo.push_back(item);
    }

    // release of list pointer
    SCRSDK:: ReleaseLensInformation (handle, list);

}
```

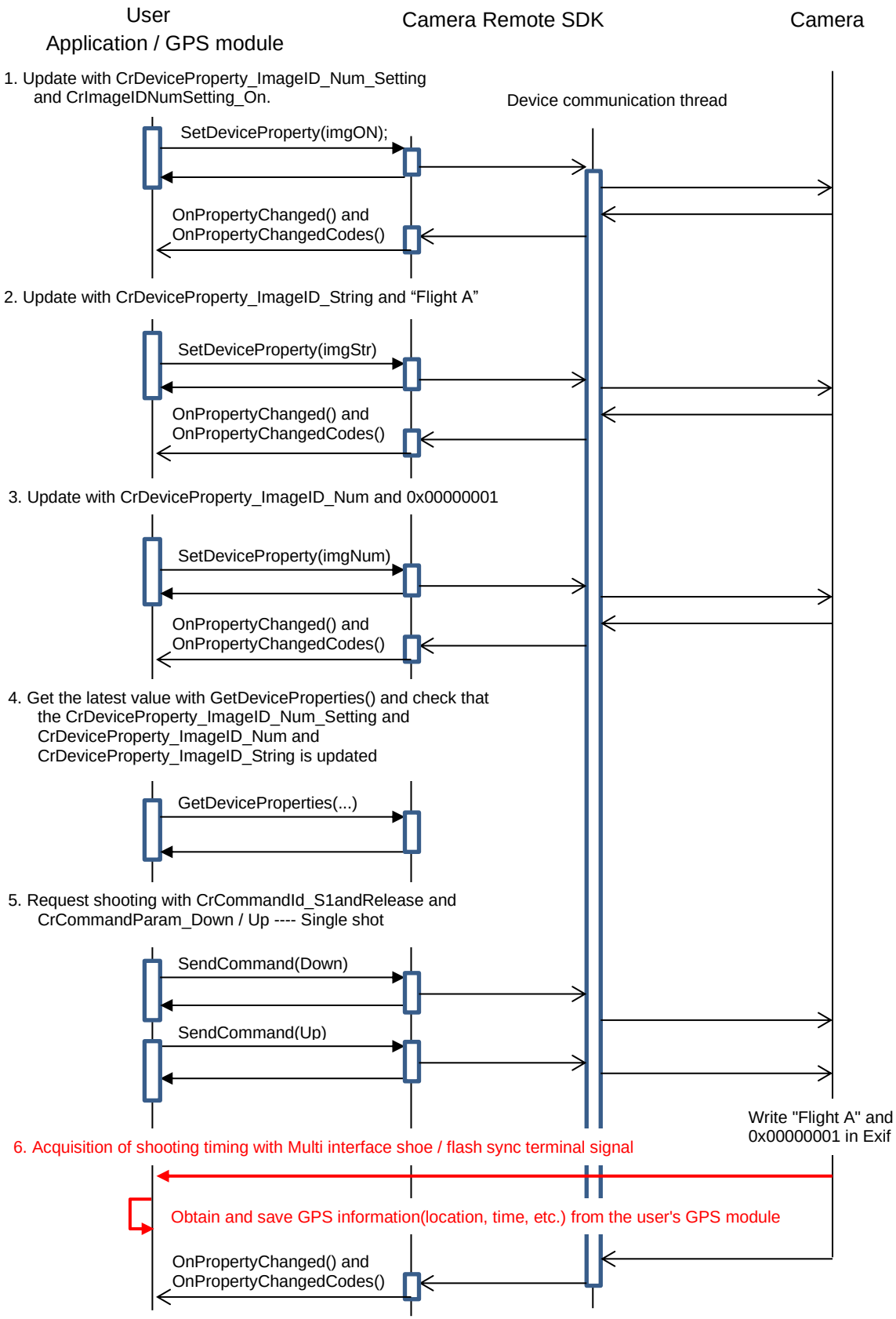
The information retrieved by GetLensInformation() can be used to know the Focus position (meter/feet).

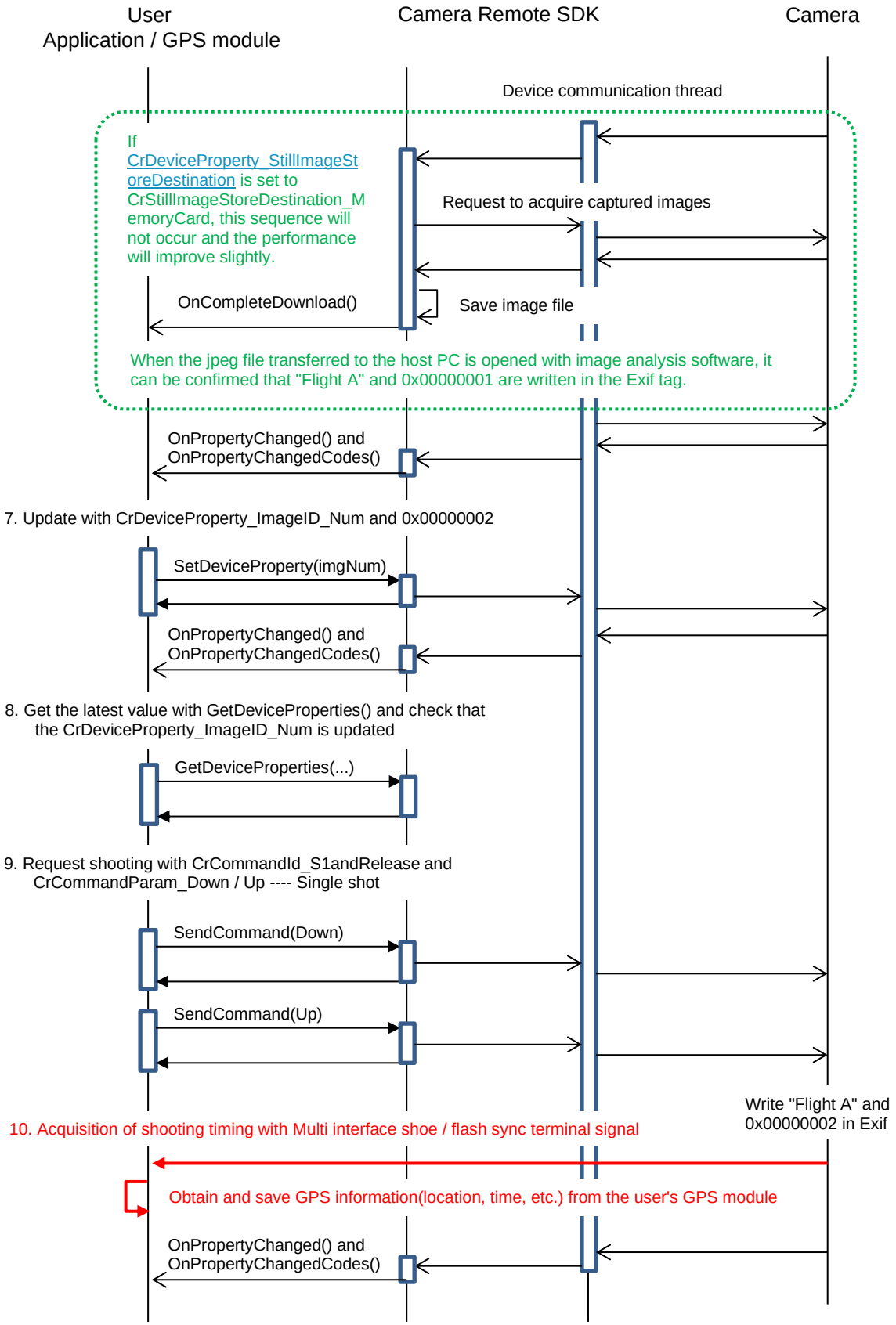
*Example:*

```
// Example of a case where CrDeviceProperty_FollowFocusPositionCurrentValue is
0x00001234
CrInt32u followVal = 0x00001234;

// If Focal distance unit is “Feet”
CrInt32u unitFeet = SCRSDK::CrLensInformationType_Feet;

for (int i=0; i < (m_lensInfo.size() - 1); ++i) {
    if (m_lensInfo[i]->type != unitFeet) continue;
    if ((m_lensInfo[i + 1]->normalizedValue <= followVal) &&
       (followVal <= m_lensInfo[i]->normalizedValue)) {
        printf(“Follow Focus Position between %d and %d\n”,
                m_lensInfo[i]->focusPosition, m_lensInfo[i + 1]->focusPosition);
        break;
    }
}
```

# GPS information and shooting image link

After shooting, if you collate the information recorded in the Exif tag of the image file with the GPS information, you can synchronize the image with the GPS information with high accuracy




User
Application / GPS module
Camera Remote SDK
Camera
Device communication thread
If CrDeviceProperty_StillImageStoreDestination is set to CrStillImageStoreDestination_MemoryCard, this sequence will not occur and the performance will improve slightly.
Request to acquire captured images
OnCompleteDownload()
Save image file
When the jpeg file transferred to the host PC is opened with image analysis software, it can be confirmed that "Flight A" and 0x00000001 are written in the Exif tag.
OnPropertyChanged() and OnPropertyChangedCodes()
7. Update with CrDeviceProperty_ImageID_Num and 0x00000002
SetDeviceProperty(imgNum)
OnPropertyChanged() and OnPropertyChangedCodes()
8. Get the latest value with GetDeviceProperties() and check that the CrDeviceProperty_ImageID_Num is updated
GetDeviceProperties(...)
9. Request shooting with CrCommandId_S1andRelease and CrCommandParam_Down / Up ---- Single shot
SendCommand(Down)
SendCommand(Up)
Write "Flight A" and 0x00000002 in Exif
10. Acquisition of shooting timing with Multi interface shoe / flash sync terminal signal
Obtain and save GPS information(location, time, etc.) from the user's GPS module
OnPropertyChanged() and OnPropertyChangedCodes()

# More information

## Trademarks and acknowledgements

Sony is a trademark or registered trademark of Sony Corporation.
All other trademarks and copyrights are the property of their respective owners