October 2022

API references for Camera Remote SDK

Camera Remote SDK API Reference

*All implied warranties, including without limitation the implied warranties of merchantability or fitness for a particular purpose, are excluded. In no event shall Sony Corporation or its licensors be liable for incidental or consequential damages of any nature, including but not limited to lost profits or commercial loss, arising out of the use of the information in this document.

SONY

First edition (February 2020)

This document is published by Sony Corporation. without any warranty*. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Sony Corporation.at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Printed versions are to be regarded as temporary reference copies only.

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
2020-02-06	1.00.00	First version
2020-06-18	1.00.01	Just SDK version proceeded with bug fix (no change in the API doc.)
2020-07-16	1.01.00	Some of DeviceProperties and Property values are added.
2020-07-28	1.02.00	“Supporting products” is updated. Some of DeviceProperties and Property values are added.
2020-08-03	1.02.00	“Supporting OS” and “Providing package” are updated.
2020-09-15	1.02.00	“Supporting products” is updated.
2020-10-15	1.02.01	Just SDK version proceeded with bug fix (no change in the API doc.) Windows version only.
2020-10-15	1.02.01	Explanation of Focus_Magnifier_Setting is updated in “CrDeviceProperty” and added in “Tips/Trouble Shooting”.
2020-12-08	1.03.00	“Supporting OS” and “Providing package” are updated. Multiple cameras can be controlled by a single SDK. Some of error codes are added.
2021-05-11	1.04.00	“Supporting products” is updated. “Supporting OS” and “Providing package” are updated. Wired LAN connection is added. Some of DeviceProperties and DeviceProperty values are added. Some of error codes are added.
2021-11-09	1.05.00	“Supporting products” is updated. Content transfer function via USB connection added. Some of callback functions are added. Some of DeviceProperties and DeviceProperty values are added. Some of error codes are added.
2021-12-07	1.05.00	"Function List" is updated for the latest version of ILCE-7RM4A and ILCE-7C.
2022-10-12	1.06.00	“Supporting products” is updated. Wired LAN connection by SSH is added. Some of callback functions are added. Some of DeviceProperties and DeviceProperty values are added. Some of error codes are added. “Function List” is updated for the latest version of ILCE-1, ILCE-7SM3 and ILCE-7M4.
2022-10-27	1.07.00	“Supporting products” is updated.

Contents

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
- DSC-RX0M2 (Ver. 3.00 or later)	https://helpguide.sony.net/dsc/1910/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

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	DSC-RX0M2 (Ver. 3.00 or later)	✓	✓	-	-
10	ILME-FX6 (Ver. 3.00 or later)	-	-	✓*	-
11	ILME-FX3 (Ver. 2.00 or later)	✓	✓	-	-
12	ILME-FX30	✓	✓	-	-

“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 8.1 64bit”, “Windows 10 64bit”, “Windows 11 64bit”

- Camera Remote SDK for Linux 64bit PC

Checked with the environment on “Ubuntu 18.04.5 LTS”, “Ubuntu 20.04.1 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 18.04.4 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 “10.15(Catalina)” and “11.1 or later(Big Sur)” and “12.1 or later(Monterey)”

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.

MENU > Network > PC Remote Function

For ILCE-1 : MENU > Network > Transfer/Remote > PC Remote Function

- Please set "PC Remote" to "On".
  For some models, the default setting of "Smartphone Connection" is "On". As it is, you cannot turn "PC Remote" into "On". Please set "Smartphone Connection" to "Off".
- The default setting of "PC Remote Cnct Method" is "USB", but if other than "USB" is set, change it to "USB".
- The menu structure of DSC-RX0M2 is different from that of ILCE model. Set "USB Connection" to "PC Remote".
  For DSC-RX0M2 : MENU > Setup3 > USB Connection

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.

MENU > Network > PC Remote Function

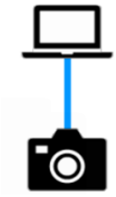
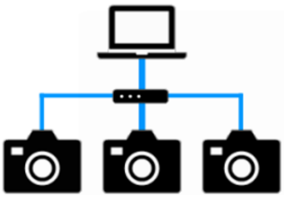
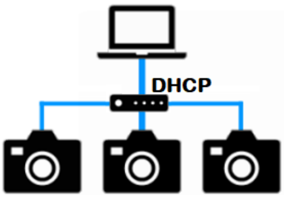
For ILCE-1 : MENU > Network > Transfer/Remote > PC Remote Function

- Please set “PC Remote” to “On”.
  For some models, the default setting of “Smartphone Connection” is “On”. As it is, you cannot turn “PC Remote” into “On”. Please set “Smartphone Connection” to “Off”.
- The default setting for “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 "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 “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".

MENU > Network > Wired LAN > IP Address Setting

- The default setting for “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.
- "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.

MENU > Network > PC Remote Function > Pairing

For ILCE-1 : MENU > Network > Transfer/Remote > PC Remote Function > Pairing

First, select “Pairing” from the camera menu 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 ILME-FX6, the following settings must be made on the camera itself.

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. Increases each time Connect() is called. Rotation with 127. See “Get the MediaProfile” 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. Ex.) Cam1(192.168.10.3) = localhost:58081 Cam2(192.168.10.4) = localhost:58082

Uninstallation

Delete all related folders and files.

When uninstalling your application which using 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.

APIs	Outline	Mode
Init	Initialize the Camera Remote SDK for use.	R/C
Release	Terminate the Camera Remote SDK.	R/C
EnumCameraObjects	Make a list of corresponding camera for the Camera Remote SDK.	R/C
CreateCameraObjectInfo	Create an ICrCameraObjectInfo object represents a Camera.	R/C
CreateCameraObjectInfoUSBConnection	Create an ICrCameraObjectInfo object that represents a camera to be connecting via USB.	R/C
CreateCameraObjectInfoEthernetConnection	Create an ICrCameraObjectInfo object that represents a camera to be connecting via Ethernet	R/C
GetFingerprint	Get the fingerprint of the camera to connect with SSH authentication	R/C
Connect	Connect to a Camera using a ICrCameraObjectInfo object before manipulation.	R/C
Disconnect	Disconnect from the Camera after use.	R/C
ReleaseDevice	Remove resources allocated with the Connect function.	R/C
GetDeviceProperties	Read camera settings.	R/C
ReleaseDeviceProperties	Release the CrDeviceProperty objects allocated by GetDeviceProperties.	R/C
SetDeviceProperty	Set camera settings.	R
SendCommand	Send control command.	R/C
GetLiveViewImage (*2)	Read the latest live-view image data from the Camera into the memory of the current machine.	R
GetLiveViewImageInfo (*2)	This function returns the size of the live-view image.	R
GetLiveViewProperties (*2)	Get live view properties from the camera.	R
ReleaseLiveViewProperties (*2)	Release the CrLiveViewProperty objects allocated by GetLiveViewProperties	R
GetDeviceSetting (*2)	This function returns the value of settings in the Camera Remote SDK.	R
SetDeviceSetting (*2)	This function modifies the value of settings in the Camera Remote SDK.	R

SetSaveInfo	This function modifies settings for saving pictures	R/C
GetSDKVersion	Get SDK version number.	R/C
GetSDKSerial	Get SDK serial number.	R/C
GetSelectDeviceProperties	Specify and read the device property from the camera.	R/C
GetSelectLiveViewProperties	Specify and read the live view property from the camera.	R
EditSDKInfo	Edit the information about the SDK stored in the config file.	R/C
GetDateFolderList (*2)	Get date folder.	C
GetContentsHandleList (*2)	Get content handle array in the date folder.	C
GetContentsDetailInfo (*2)	Get contents Information.	C
ReleaseDateFolderList (*2)	Release the CrMtpFolderInfo objects allocated by GetDateFolderList.	C
ReleaseContentsHandleList (*2)	Release the CrMtpContentsInfo object allocated by GetContentsHandleList.	C
PullContentsFile (*1*2)	Get(download) contents file.	C
GetContentsThumbnailImage (*2)	Read thumbnail image data into the memory of the current machine.	C
DownloadSettingFile (*2)	Get(download) the camera settings file.	R
UploadSettingFile (*2)	Update(upload) the camera settings file.	R
RequestDisplayStringList	Request a list of display menu strings	R
GetDisplayStringTypes	Get referenceable display menu string type.	R
GetDisplayStringList	Get the list of display menu strings.	R
ReleaseDisplayStringTypes	Release the CrDisplayStringType objects allocated by GetDisplayStringTypes.	R
ReleaseDisplayStringList	Release the CrDisplayStringListInfo objects allocated by GetDisplayStringList.	R
GetMediaProfile (*3)	Get the MediaProfile Lists.	R
ReleaseMediaProfile (*3)	Release the MediaProfile Lists.	R
RequestLensInformation	Request the acquisition of Lens information.	R
GetLensInformation	Get the Lens information Lists.	R
ReleaseLensInformation	Release the Lens information Lists.	R

*1 : Large files may not be handled depending on the OS.
*2 : Not supported for ILME-FX6.
*3 : Supported only for ILME-FX6.
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.
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.

(1/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Shutter Half Release	CrDeviceProperty_S1	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Shutter Release	CrCommandId_Release	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
AELock Indication	CrDeviceProperty_AEL	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
FEL Lock Indication	CrDeviceProperty_FEL	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
AWBLock Indication	CrDeviceProperty_AWBL	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
F-Number	CrDeviceProperty_FNumber	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Exposure Bias Compensation	CrDeviceProperty_ExposureBiasCompensation	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Shutter Speed	CrDeviceProperty_ShutterSpeed	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
ISO Sensitivity	CrDeviceProperty_IsoSensitivity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Focus Area	CrDeviceProperty_FocusArea	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Exposure Program Mode	CrDeviceProperty_ExposureProgramMode	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Compress File Format(Still)	CrDeviceProperty_CompressionFileFormatStill	✓	-	✓	-	-	✓	✓	-	-	-	✓	✓
File Format(Still)	CrDeviceProperty_FileType	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Media SLOT1 File Format(Still)	CrDeviceProperty_MediaSLOT1_FileType	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT2 File Format(Still)	CrDeviceProperty_MediaSLOT2_FileType	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
JPEG Quality	CrDeviceProperty_JpegQuality	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Media SLOT1 JPEG Quality	CrDeviceProperty_MediaSLOT1_JpegQuality	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT2 JPEG Quality	CrDeviceProperty_MediaSLOT2_JpegQuality	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
White Balance	CrDeviceProperty_WhiteBalance	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Focus Mode	CrDeviceProperty_FocusMode	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Exposure Metering Mode	CrDeviceProperty_MeteringMode	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Flash Mode	CrDeviceProperty_FlashMode	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Flash Compensation	CrDeviceProperty_FlashCompensation	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Wireless Flash Setting	CrDeviceProperty_WirelessFlash	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Red Eye Reduction	CrDeviceProperty_RedEyeReduction	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Still Capture Mode	CrDeviceProperty_DriveMode	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Dynamic Range Optimizer	CrDeviceProperty_DRO	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Image Size	CrDeviceProperty_ImageSize	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Media SLOT1 Image Size	CrDeviceProperty_MediaSLOT1_ImageSize	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT2 Image Size	CrDeviceProperty_MediaSLOT2_ImageSize	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Aspect Ratio	CrDeviceProperty_AspectRatio	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Picture Effect	CrDeviceProperty_PictureEffect	-	✓	-	✓	✓	-	-	✓	✓	-	-	-
Color Temperature	CrDeviceProperty_Colortemp	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Biaxial Fine Tuning A-B Direction	CrDeviceProperty_ColorTuningAB	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Biaxial Fine Tuning G-M Direction	CrDeviceProperty_ColorTuningGM	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓

(2/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Live View Display Effect	CrDeviceProperty_LiveViewDisplayEffect	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Still Image Save Destination	CrDeviceProperty_StillImageStoreDestination	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Position Key Setting	CrDeviceProperty_PriorityKeySettings	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Focus Magnifier Setting	CrDeviceProperty_Focus_Magnifier_Setting	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Date/Time Setting	CrDeviceProperty_DateTime_Settings	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Focus Near/Far Setting	CrDeviceProperty_NearFar	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Live View Image Quality	CrDeviceProperty_LiveView_Image_Quality	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Interval REC Mode	CrDeviceProperty_Interval_Rec_Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Still Image Trans Size	CrDeviceProperty_Still_Image_Trans_Size	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
RAW+J PC Save Image	CrDeviceProperty_RAW_J_PC_Save_Image	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capture Standby	CrDeviceProperty_CustomWB_Capture_Standby	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capture Standby Cancel	CrDeviceProperty_CustomWB_Capture_Standby_Cancel	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capture	CrDeviceProperty_CustomWB_Capture	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Shooting File Info	CrDeviceProperty_SnapshotInfo	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Battery Remaining	CrDeviceProperty_BatteryRemain	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
Battery Level Indicator	CrDeviceProperty_BatteryLevel	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Movie Recording State	CrDeviceProperty_RecordingState	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
LiveView Status	CrDeviceProperty_LiveViewStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Focus Indication	CrDeviceProperty_FocusIndication	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Media SLOT1 Status	CrDeviceProperty_MediaSLOT1_Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media SLOT1 Remaining number shots	CrDeviceProperty_MediaSLOT1_RemainingNumber	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Media SLOT1 Remaining shooting time	CrDeviceProperty_MediaSLOT1_RemainingTime	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media SLOT1 Full Format Enable Status	CrDeviceProperty_MediaSLOT1_FormatEnableStatus	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Media SLOT1 Quick Format Enable Status	CrDeviceProperty_MediaSLOT1_QuickFormatEnableStatus	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT2 Status	CrDeviceProperty_MediaSLOT2_Status	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓
Media SLOT2 Remaining number shots	CrDeviceProperty_MediaSLOT2_RemainingNumber	✓	✓	✓	✓	✓	✓	✓	-	-	-	✓	✓
Media SLOT2 Remaining shooting time	CrDeviceProperty_MediaSLOT2_RemainingTime	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓
Media SLOT2 Full Format Enable Status	CrDeviceProperty_MediaSLOT2_FormatEnableStatus	✓	✓	✓	✓	-	✓	✓	-	-	-	✓	✓
Media SLOT2 Quick Format Enable Status	CrDeviceProperty_MediaSLOT2_QuickFormatEnableStatus	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media Format Progress Rate	CrDeviceProperty_Media_FormatProgressRate	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Execute Format the Media	CrCommandId_MediaFormat	✓	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Execute Quick Format the Media	CrCommandId_MediaQuickFormat	✓	-	✓	-	-	-	✓	-	-	-	✓	✓

(3/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
AF Area Position	CrDeviceProperty_AF_Area_Position	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Zoom Scale	CrDeviceProperty_Zoom_Scale	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Zoom Setting	CrDeviceProperty_Zoom_Setting	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Zoom Operation	CrDeviceProperty_Zoom_Operation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
File Format(Movie)	CrDeviceProperty_Movie_File_Format	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Recording Setting(Movie)	CrDeviceProperty_Movie_Recording_Setting	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Recording Frame Rate Setting(Movie)	CrDeviceProperty_Movie_Recording_FrameRateSetting	-	-	✓	-	-	✓	✓	-	-	✓	✓	✓
Interval REC Status	CrDeviceProperty_Interval_Rec_Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Control Movie Rec button	CrCommandId_MovieRecord	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Custom WB Execution State	CrDeviceProperty_CustomWB_Execution_State	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capturable Area	CrDeviceProperty_CustomWB_Capturable_Area	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capture Frame Size	CrDeviceProperty_CustomWB_Capture_Frame_Size	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Custom WB Capture Operation Enable Status	CrDeviceProperty_CustomWB_Capture_Operation	✓	✓	✓	✓	✓	✓	✓	✓	-	-	✓	✓
Zoom Operation Enable Status	CrDeviceProperty_Zoom_Operation_Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zoom Bar Information	CrDeviceProperty_Zoom_Bar_Information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zoom Type Status	CrDeviceProperty_Zoom_Type_Status	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
RAW File Compression Type	CrDeviceProperty_RAW_FileCompressionType	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT1 RAW File Compression Type	CrDeviceProperty_MediaSLOT1_RAW_FileCompressionType	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Media SLOT2 RAW File Compression Type	CrDeviceProperty_MediaSLOT2_RAW_FileCompressionType	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Cancel Media Format Enable Status	CrDeviceProperty_Cancel_Media_FormatEnableStatus	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Cancel media format	CrCommandId_CancelMediaFormat	✓	-	✓	-	-	-	✓	-	-	-	✓	✓
Shutter Half Release and Release	CrCommandId_S1andRelease	✓	✓	✓	-	-	✓	✓	✓	-	-	✓	✓
Save Zoom&FocusPosition in presets	CrDeviceProperty_ZoomAndFocusPosition_Save	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Load Zoom&FocusPosition from presets	CrDeviceProperty_ZoomAndFocusPosition_Load	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Remocon Zoom Speed Type	CrDeviceProperty_Remocon_Zoom_Speed_Type	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Zoom Speed Range	CrDeviceProperty_Zoom_Speed_Range	-	-	✓	-	-	-	✓	-	-	-	-	✓
Sdk Control Mode	CrDeviceProperty_SdkControlMode	✓	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Get content accessibility status	CrDeviceProperty_ContentsTransferStatus	✓	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Cancel Content transfer Enable Status	CrDeviceProperty_ContentsTransferCancelEnableStatus	✓	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Content transfer Progress	CrDeviceProperty_ContentsTransferProgress	✓	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓
Cancel Contents transfer	CrCommandId_CancelContentsTransfer	✓	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓

(4/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Iris Mode Setting	CrDeviceProperty_IrisModeSetting	-	-	✓	-	-	-	-	-	-	✓	✓	✓
Shutter Mode Setting	CrDeviceProperty_ShutterModeSetting	-	-	✓	-	-	-	-	-	-	✓	✓	✓
Gain Control Setting	CrDeviceProperty_GainControlSetting	-	-	✓	-	-	-	-	-	-	✓	✓	✓
Gain Base Iso Sensitivity	CrDeviceProperty_GainBaseIsoSensitivity	-	-	-	-	-	-	-	-	-	✓	✓	✓
Gain Base Sensitivity	CrDeviceProperty_GainBaseSensitivity	-	-	-	-	-	-	-	-	-	✓	-	-
Exposure Index	CrDeviceProperty_ExposureIndex	-	-	-	-	-	-	-	-	-	✓	✓	✓
BaseLook Value	CrDeviceProperty_BaseLookValue	-	-	-	-	-	-	-	-	-	✓	✓	✓
Playback Media	CrDeviceProperty_PlaybackMedia	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Monitor DISP(Screen Display) Mode Candidate	CrDeviceProperty_DispModeCandidate	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Monitor DISP(Screen Display) Mode Setting	CrDeviceProperty_DispModeSetting	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Monitor DISP(Screen Display) Mode	CrDeviceProperty_DispMode	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Touch Operation	CrDeviceProperty_TouchOperation	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Select Finder/Monitor	CrDeviceProperty_SelectFinder	-	-	✓	-	-	-	✓	-	-	-	-	-
Auto Power OFF Temperature	CrDeviceProperty_AutoPowerOffTemperature	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Body Key Lock	CrDeviceProperty_BodyKeyLock	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Image ID(Numerical Value) Setting	CrDeviceProperty_ImageID_Num_Setting	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Image ID(Numerical Value)	CrDeviceProperty_ImageID_Num	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Image ID(String)	CrDeviceProperty_ImageID_String	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Exposure Control Mode	CrDeviceProperty_ExposureCtrlType	-	-	✓	-	-	-	-	-	-	-	✓	✓
Monitor LUT Setting(All Line)	CrDeviceProperty_MonitorLUTSetting	-	-	-	-	-	-	-	-	-	-	✓	✓
ISO Current Sensitivity	CrDeviceProperty_IsoCurrentSensitivity	-	-	✓	-	-	-	-	-	-	✓	✓	✓
Camera-Setting Save Operation Enable Status	CrDeviceProperty_CameraSetting_SaveOperationEnableStatus	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Camera-Setting Read Operation Enable Status	CrDeviceProperty_CameraSetting_ReadOperationEnableStatus	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Camera-Setting Save/Read State	CrDeviceProperty_CameraSetting_SaveRead_State	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Camera Setting Reset Enable Status	CrDeviceProperty_CameraSettingsResetEnableStatus	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Execute Camera Setting Reset	CrCommandId_CameraSettingsReset	-	-	✓	-	-	-	✓	-	-	-	✓	✓
APS-C or Full Switching Setting	CrDeviceProperty_APS_C_or_Full_SwitchingSetting	✓	-	✓	-	-	-	✓	-	-	-	✓	-
APS-C or Full Switching Enable Status	CrDeviceProperty_APS_C_or_Full_SwitchingEnableStatus	✓	-	✓	-	-	-	✓	-	-	-	✓	-
Execute APS-C or Full Switching	CrCommandId_APS_C_or_Full_Switching	✓	-	✓	-	-	-	✓	-	-	-	✓	-
Execute Movie Rec Button (2nd)	CrCommandId_MovieRecButtonToggle	-	-	-	-	-	-	-	-	-	✓	-	-
Execute Cancel Remote Touch Operation	CrCommandId_CancelRemoteTouchOperation	-	-	-	-	-	-	-	-	-	✓	-	-

(5/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Focal Distance in Meter	CrDeviceProperty_FocalDistanceInMeter	-	-	-	-	-	-	-	-	-	✓	-	-
Focal Distance in Feet	CrDeviceProperty_FocalDistanceInFeet	-	-	-	-	-	-	-	-	-	✓	-	-
Focal Distance Unit Setting	CrDeviceProperty_FocalDistanceUnitSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Digital Zoom Scale	CrDeviceProperty_DigitalZoomScale	-	-	-	-	-	-	-	-	-	✓	-	-
Zoom Distance	CrDeviceProperty_ZoomDistance	-	-	-	-	-	-	-	-	-	✓	-	-
Get/Set the Zoom Distance Unit Setting	CrDeviceProperty_ZoomDistanceUnitSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Mode Status	CrDeviceProperty_ShutterModeStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Slow	CrDeviceProperty_ShutterSlow	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Slow Frames	CrDeviceProperty_ShutterSlowFrames	-	-	-	-	-	-	-	-	-	✓	-	-
Recording Resolution For Main(Movie)	CrDeviceProperty_Movie_Recording_ResolutionForMain	-	-	-	-	-	-	-	-	-	✓	-	-
Recording Resolution For Proxy(Movie)	CrDeviceProperty_Movie_Recording_ResolutionForProxy	-	-	-	-	-	-	-	-	-	✓	-	-
Recording Frame Rate Proxy Setting(Movie)	CrDeviceProperty_Movie_Recording_FrameRateProxySetting	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Shooting Mode	CrDeviceProperty_MovieShootingMode	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Shooting Mode Color Gamut	CrDeviceProperty_MovieShootingModeColorGamut	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Shooting Mode Target Display	CrDeviceProperty_MovieShootingModeTargetDisplay	-	-	-	-	-	-	-	-	-	✓	-	-
Depth of Field Adjustment Mode	CrDeviceProperty_DepthOfFieldAdjustmentMode	-	-	-	-	-	-	-	-	-	✓	-	-
Depth of Field Adjustment Interlocking Mode State	CrDeviceProperty_DepthOfFieldAdjustmentInterlockingMode	-	-	-	-	-	-	-	-	-	✓	-	-
Color Temperature	CrDeviceProperty_ColortempStep	-	-	-	-	-	-	-	-	-	✓	-	-
White Balance Mode Setting	CrDeviceProperty_WhiteBalanceModeSetting	-	-	-	-	-	-	-	-	-	✓	-	-
White Balance Tint	CrDeviceProperty_WhiteBalanceTint	-	-	-	-	-	-	-	-	-	✓	-	-
White Balance Tint Step	CrDeviceProperty_WhiteBalanceTintStep	-	-	-	-	-	-	-	-	-	✓	-	-
Execute the Focus Operation	CrDeviceProperty_Focus_Operation	-	-	-	-	-	-	-	-	-	✓	-	-
Focus Speed Range	CrDeviceProperty_Focus_Speed_Range	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter ECS Setting	CrDeviceProperty_ShutterECSSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter ECS Number	CrDeviceProperty_ShutterECSNumber	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter ECS Number Step	CrDeviceProperty_ShutterECSNumberStep	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter ECS Frequency	CrDeviceProperty_ShutterECSFrequency	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 1	CrDeviceProperty_ButtonAssignmentAssignable1	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 2	CrDeviceProperty_ButtonAssignmentAssignable2	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 3	CrDeviceProperty_ButtonAssignmentAssignable3	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 4	CrDeviceProperty_ButtonAssignmentAssignable4	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 5	CrDeviceProperty_ButtonAssignmentAssignable5	-	-	-	-	-	-	-	-	-	✓	-	-

(6/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Button Assignment Assignable 6	CrDeviceProperty_ButtonAssignmentAssignable6	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 7	CrDeviceProperty_ButtonAssignmentAssignable7	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 8	CrDeviceProperty_ButtonAssignmentAssignable8	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment Assignable 9	CrDeviceProperty_ButtonAssignmentAssignable9	-	-	-	-	-	-	-	-	-	✓	-	-
Button Assignment LensAssignable 1	CrDeviceProperty_ButtonAssignmentLensAssignable1	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 1	CrDeviceProperty_AssignableButton1	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 2	CrDeviceProperty_AssignableButton2	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 3	CrDeviceProperty_AssignableButton3	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 4	CrDeviceProperty_AssignableButton4	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 5	CrDeviceProperty_AssignableButton5	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 6	CrDeviceProperty_AssignableButton6	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 7	CrDeviceProperty_AssignableButton7	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 8	CrDeviceProperty_AssignableButton8	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button 9	CrDeviceProperty_AssignableButton9	-	-	-	-	-	-	-	-	-	✓	-	-
LensAssignable Button 1	CrDeviceProperty_LensAssignableButton1	-	-	-	-	-	-	-	-	-	✓	-	-
Focus Mode Setting	CrDeviceProperty_FocusModeSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Angle	CrDeviceProperty_ShutterAngle	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Setting	CrDeviceProperty_ShutterSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Mode	CrDeviceProperty_ShutterMode	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Speed Value	CrDeviceProperty_ShutterSpeedValue	-	-	-	-	-	-	-	-	-	✓	-	-
Shutter Speed Current Value	CrDeviceProperty_ShutterSpeedCurrentValue	-	-	-	-	-	-	-	-	-	✓	-	-
ND Filter	CrDeviceProperty_NDFilter	-	-	-	-	-	-	-	-	-	✓	-	-
ND Filter Mode	CrDeviceProperty_NDFilterMode	-	-	-	-	-	-	-	-	-	✓	-	-
ND Filter Mode Setting	CrDeviceProperty_NDFilterModeSetting	-	-	-	-	-	-	-	-	-	✓	-	-
ND Filter Value	CrDeviceProperty_NDFilterValue	-	-	-	-	-	-	-	-	-	✓	-	-
Gain Unit Setting	CrDeviceProperty_GainUnitSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Gain dB Value	CrDeviceProperty_GaindBValue	-	-	-	-	-	-	-	-	-	✓	-	-
Gain dB Current Value	CrDeviceProperty_GaindBCurrentValue	-	-	-	-	-	-	-	-	-	✓	-	-
AWB	CrDeviceProperty_AWB	-	-	-	-	-	-	-	-	-	✓	-	-
SceneFile Index	CrDeviceProperty_SceneFileIndex	-	-	-	-	-	-	-	-	-	✓	-	-
Current SceneFile Edited Info	CrDeviceProperty_CurrentSceneFileEdited	-	-	-	-	-	-	-	-	-	✓	-	-

(7/8)

Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Movie Play button	CrDeviceProperty_MoviePlayButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Play Pause button	CrDeviceProperty_MoviePlayPauseButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Play Stop button	CrDeviceProperty_MoviePlayStopButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Forward button	CrDeviceProperty_MovieForwardButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Rewind button	CrDeviceProperty_MovieRewindButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Next button	CrDeviceProperty_MovieNextButton	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Prev button	CrDeviceProperty_MoviePrevButton	-	-	-	-	-	-	-	-	-	✓	-	-
RecReview button	CrDeviceProperty_MovieRecReviewButton	-	-	-	-	-	-	-	-	-	✓	-	-
Face Eye Detection AF	CrDeviceProperty_FaceEyeDetectionAF	-	-	-	-	-	-	-	-	-	✓	-	-
AF Transition Speed	CrDeviceProperty_AFTransitionSpeed	-	-	-	-	-	-	-	-	-	✓	-	-
AF Subj Shift Sens	CrDeviceProperty_AFSubjShiftSens	-	-	-	-	-	-	-	-	-	✓	-	-
AF Assist	CrDeviceProperty_AFAssist	-	-	-	-	-	-	-	-	-	✓	-	-
ND PRESET or VARIABLE Switching Setting	CrDeviceProperty_NDPresetOrVariableSwitchingSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Function of Remote Touch Operation	CrDeviceProperty_FunctionOfRemoteTouchOperation	-	-	-	-	-	-	-	-	-	✓	-	-
Execute Remote Touch Operation(x,y)	CrDeviceProperty_RemoteTouchOperation	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Playing State	CrDeviceProperty_MoviePlayingState	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Playing Speed	CrDeviceProperty_MoviePlayingSpeed	-	-	-	-	-	-	-	-	-	✓	-	-
Media SLOT1 Player	CrDeviceProperty_MediaSLOT1Player	-	-	-	-	-	-	-	-	-	✓	-	-
Media SLOT2 Player	CrDeviceProperty_MediaSLOT2Player	-	-	-	-	-	-	-	-	-	✓	-	-
Get/Set the Battery Remain Display Unit	CrDeviceProperty_BatteryRemainDisplayUnit	-	-	-	-	-	-	-	-	-	✓	-	-
Battery Remaining in minutes	CrDeviceProperty_BatteryRemainingInMinutes	-	-	-	-	-	-	-	-	-	✓	-	-
Battery Remaining in voltage	CrDeviceProperty_BatteryRemainingInVoltage	-	-	-	-	-	-	-	-	-	✓	-	-
Power Source	CrDeviceProperty_PowerSource	-	-	-	-	-	-	-	-	-	✓	-	-
DC voltage	CrDeviceProperty_DCVoltage	-	-	-	-	-	-	-	-	-	✓	-	-
Focus TouchSpot Status	CrDeviceProperty_FocusTouchSpotStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Focus Tracking Status	CrDeviceProperty_FocusTrackingStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Clip Name Create by The Next Rec.	CrDeviceProperty_RecorderClipName	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Control Main Setting	CrDeviceProperty_RecorderControlMainSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Control Proxy Setting	CrDeviceProperty_RecorderControlProxySetting	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Start Main	CrDeviceProperty_RecorderStartMain	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Start Proxy	CrDeviceProperty_RecorderStartProxy	-	-	-	-	-	-	-	-	-	✓	-	-

(8/8)

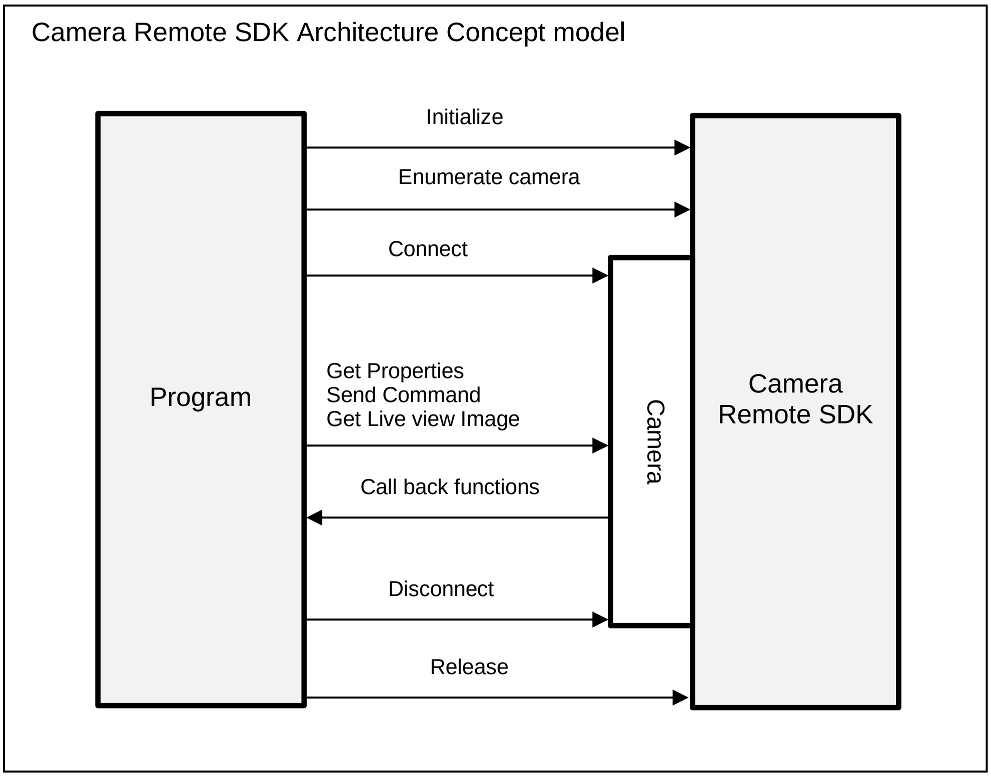
Functions	DeviceProperty Code / Command Id	ILCE-1	ILCE-9M2	ILCE-7RM5	ILCE-7RM4A	ILCE-7RM4	ILCE-7SM3	ILCE-7M4	ILCE-7C	DSC-RX0M2	ILME-FX6	ILME-FX3	ILME-FX30
Recorder Main Status	CrDeviceProperty_RecorderMainStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Proxy Status	CrDeviceProperty_RecorderProxyStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Recorder Ext Raw Status	CrDeviceProperty_RecorderExtRawStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Information of Recorder Save Destination	CrDeviceProperty_RecorderSaveDestination	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 1	CrDeviceProperty_AssignableButtonIndicator1	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 2	CrDeviceProperty_AssignableButtonIndicator2	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 3	CrDeviceProperty_AssignableButtonIndicator3	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 4	CrDeviceProperty_AssignableButtonIndicator4	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 5	CrDeviceProperty_AssignableButtonIndicator5	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 6	CrDeviceProperty_AssignableButtonIndicator6	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 7	CrDeviceProperty_AssignableButtonIndicator7	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 8	CrDeviceProperty_AssignableButtonIndicator8	-	-	-	-	-	-	-	-	-	✓	-	-
Assignable Button Indicator 9	CrDeviceProperty_AssignableButtonIndicator9	-	-	-	-	-	-	-	-	-	✓	-	-
LensAssignable Button Indicator 1	CrDeviceProperty_LensAssignableButtonIndicator1	-	-	-	-	-	-	-	-	-	✓	-	-
Software Version	CrDeviceProperty_SoftwareVersion	-	-	-	-	-	-	-	-	-	✓	-	-
Movie Rec Button (Toggle) Enable Status	CrDeviceProperty_MovieRecButtonToggleEnableStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Remote Touch Operation Enable Status	CrDeviceProperty_RemoteTouchOperationEnableStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Cancel Remote Touch Operation Enable Status	CrDeviceProperty_CancelRemoteTouchOperationEnableStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Lens Information Enable Status	CrDeviceProperty_LensInformationEnableStatus	-	-	-	-	-	-	-	-	-	✓	-	-
Follow Focus Position	CrDeviceProperty_FollowFocusPositionSetting	-	-	-	-	-	-	-	-	-	✓	-	-
Follow Focus Position Current Value	CrDeviceProperty_FollowFocusPositionCurrentValue	-	-	-	-	-	-	-	-	-	✓	-	-
Focus Bracket Shot Number	CrDeviceProperty_FocusBracketShotNumber	-	-	✓	-	-	-	-	-	-	-	-	-
Focus Bracket Focus Range	CrDeviceProperty_FocusBracketFocusRange	-	-	✓	-	-	-	-	-	-	-	-	-
Focus Bracket Shooting Status	CrDeviceProperty_FocusBracketShootingStatus	-	-	✓	-	-	-	-	-	-	-	-	-

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.

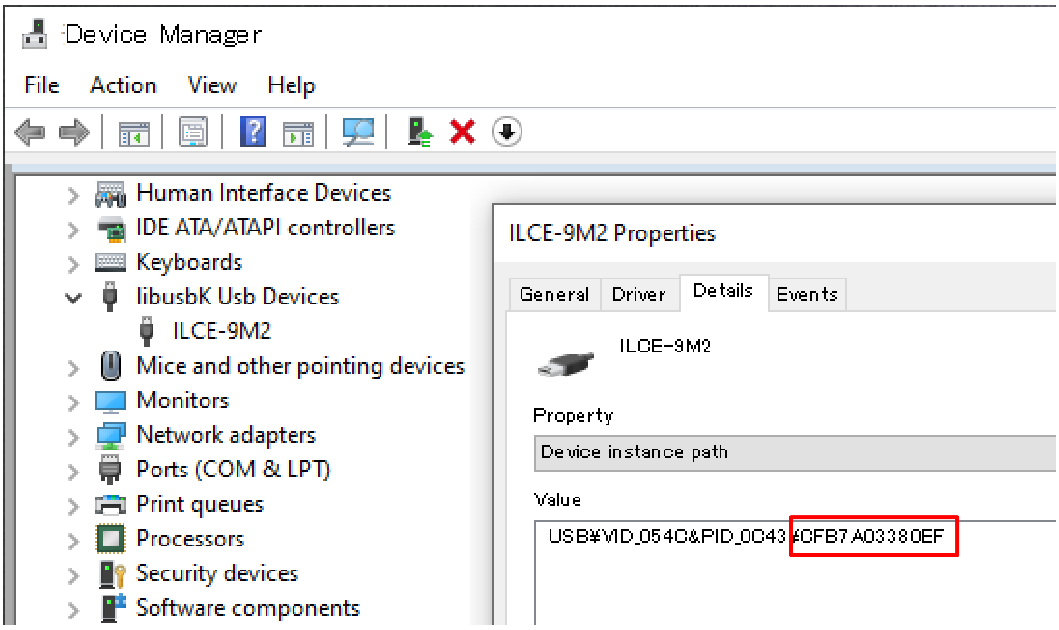
Use CreateCameraObjectInfoUSBConnection() for a USB connection and CreateCameraObjectInfoEthernetConnection() for an Ethernet connection.
The reason why we have prepared for USB connection and Ethernet connection respectively is that the required conditions differ depending on the connection method.

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



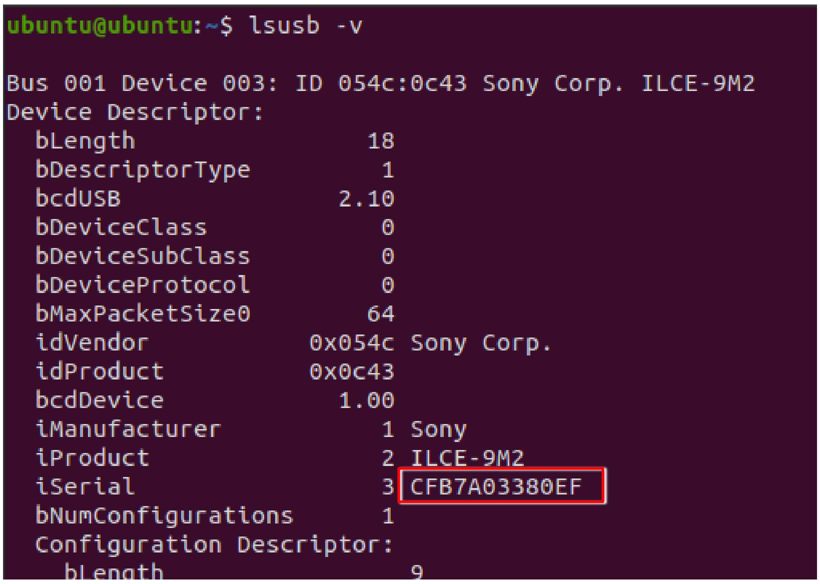
Property
Device instance path

Value
USB¥VID_054C&PID_0C43¥CFB7A03380EF

- 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

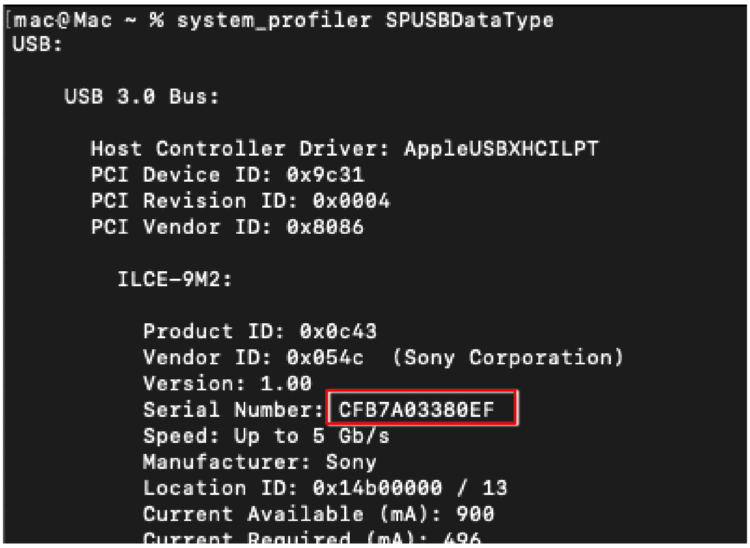


```
ubuntu@ubuntu:~$ lsusb -v

Bus 001 Device 003: ID 054c:0c43 Sony Corp. ILCE-9M2
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.10
  bDeviceClass            0
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        64
  idVendor           0x054c Sony Corp.
  idProduct          0x0c43
  bcdDevice            1.00
  iManufacturer           1 Sony
  iProduct                2 ILCE-9M2
  iSerial                 3 CFB7A03380EF
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
```

- 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



```
mac@Mac ~ % system_profiler SPUSBDataType
USB:

    USB 3.0 Bus:

      Host Controller Driver: AppleUSBXHCILPT
      PCI Device ID: 0x9c31
      PCI Revision ID: 0x0004
      PCI Vendor ID: 0x8086

        ILCE-9M2:

          Product ID: 0x0c43
          Vendor ID: 0x054c  (Sony Corporation)
          Version: 1.00
          Serial Number: CFB7A03380EF
          Speed: Up to 5 Gb/s
          Manufacturer: Sony
          Location ID: 0x14b00000 / 13
          Current Available (mA): 900
          Current Required (mA): 496
```

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, and the third parameter specifies the IP Address of the camera to connect to, forth 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.

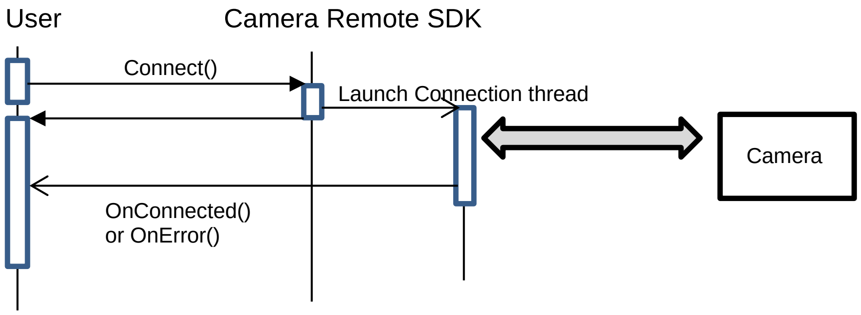
The following is an example of connecting to an SSH certified camera.

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.

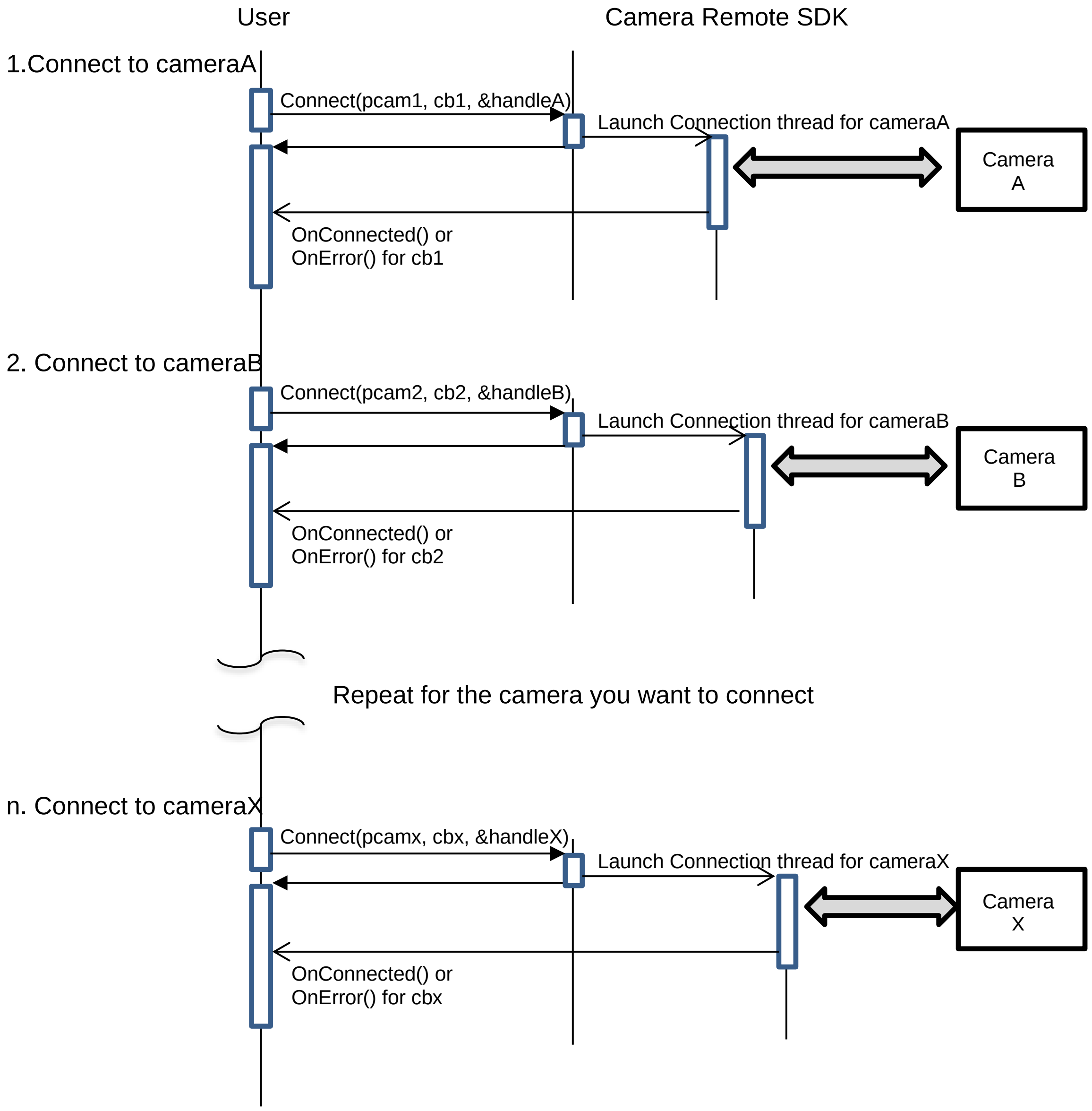
No.	User operation	Physical (USB)	Camera Remote SDK				
			Connection status with the camera				
				DeviceHandle	Camera communication thread		
						Main Loop （＊１）	Sub Loop （＊２）
Case 1	Connect/Disconnect transition						
1	Connect the camera to the PC	Disconnected -> Connected	-	-	-		
2	Call Connect()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

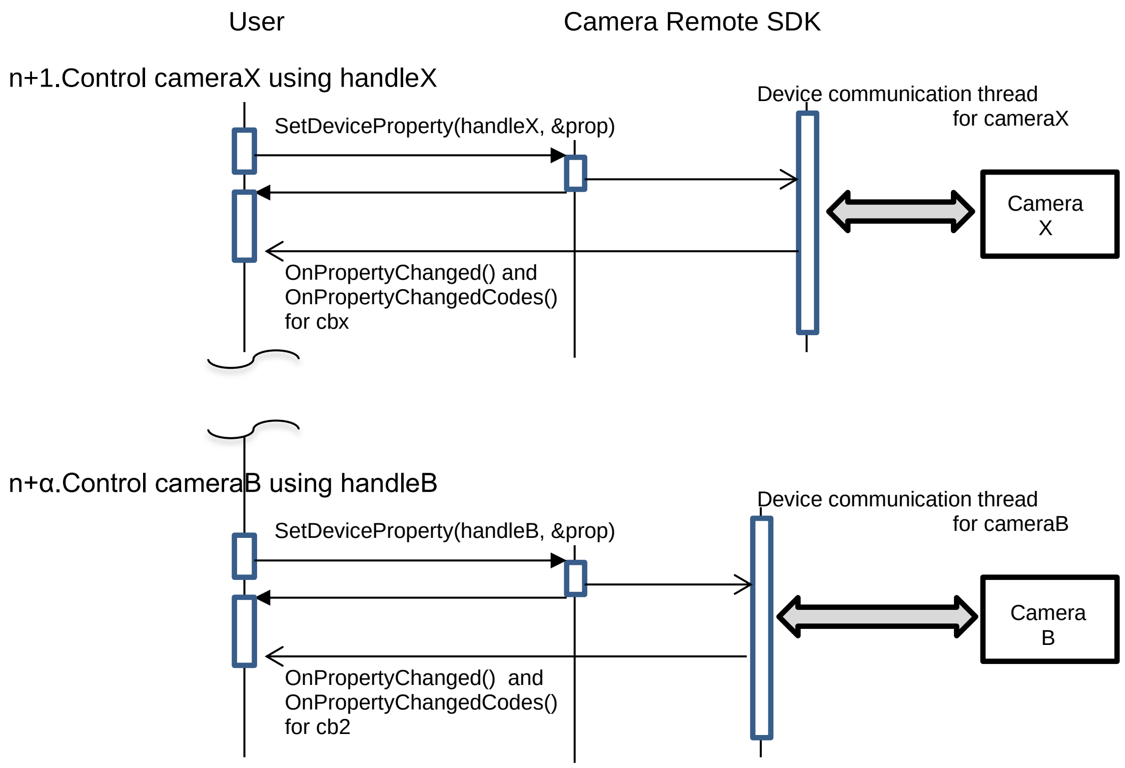
Note : If CrReconnecting_OFF is specified for the fourth parameter of the Connect(), automatic reconnection will not be performed in all cases.

Connect/Disconnect multiple cameras

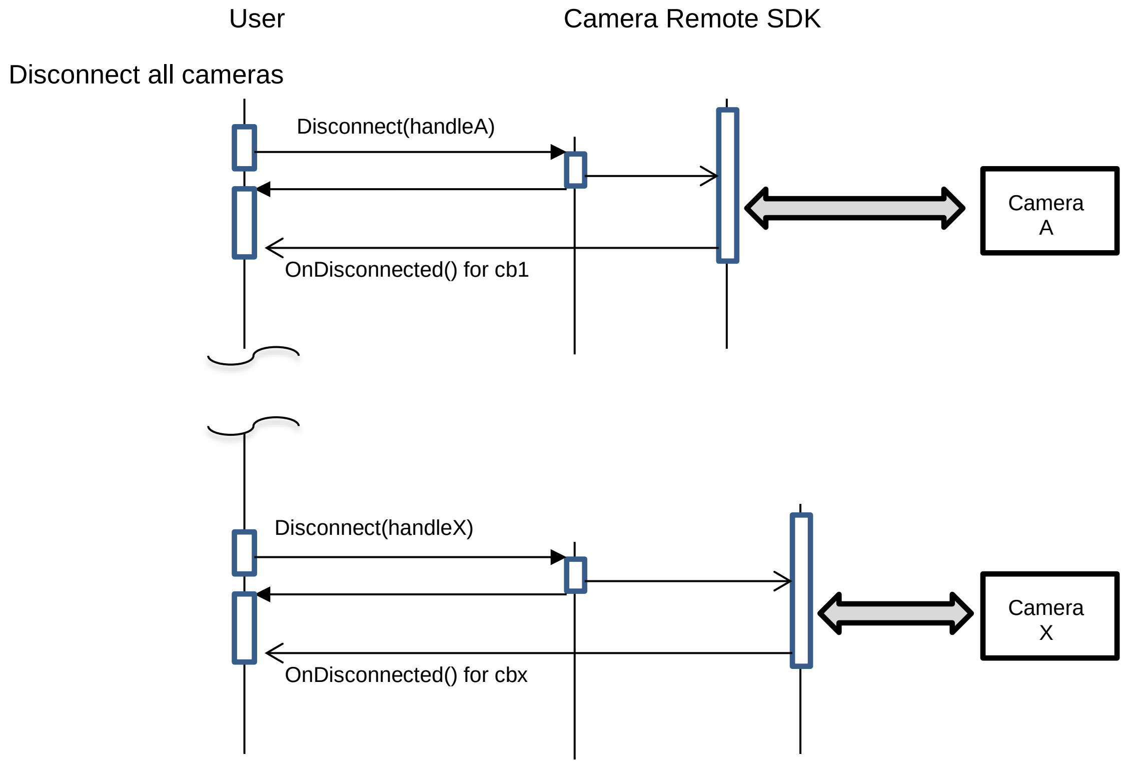
To control multiple cameras, call the Connect() function for the number of cameras and get a handle for the number of cameras.
With each handle you get, you can control each camera.


User
Camera Remote SDK
1.Connect to cameraA
Connect(pcam1, cb1, &handleA)
Launch Connection thread for cameraA
Camera A
OnConnected() or OnError() for cb1
2. Connect to cameraB
Connect(pcam2, cb2, &handleB)
Launch Connection thread for cameraB
Camera B
OnConnected() or OnError() for cb2
Repeat for the camera you want to connect
n. Connect to cameraX
Connect(pcamx, cbx, &handleX)
Launch Connection thread for cameraX
Camera X
OnConnected() or OnError() for cbx

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


User
Camera Remote SDK
n+1.Control cameraX using handleX
Device communication thread
for cameraX
SetDeviceProperty(handleX, &prop)
Camera
X
OnPropertyChanged() and
OnPropertyChangedCodes()
for cbx
n+α.Control cameraB using handleB
Device communication thread
for cameraB
SetDeviceProperty(handleB, &prop)
Camera
B
OnPropertyChanged() and
OnPropertyChangedCodes()
for cb2

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.
  Ex.1: MENU > Setup4 > USB Power Supply
  Ex.2: MENU > Setup > USB > USB Power Supply

- 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++) {
                        switch (pProperties[n].code) {
                        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].code) {
                                    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 CrLiveViewPropety 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

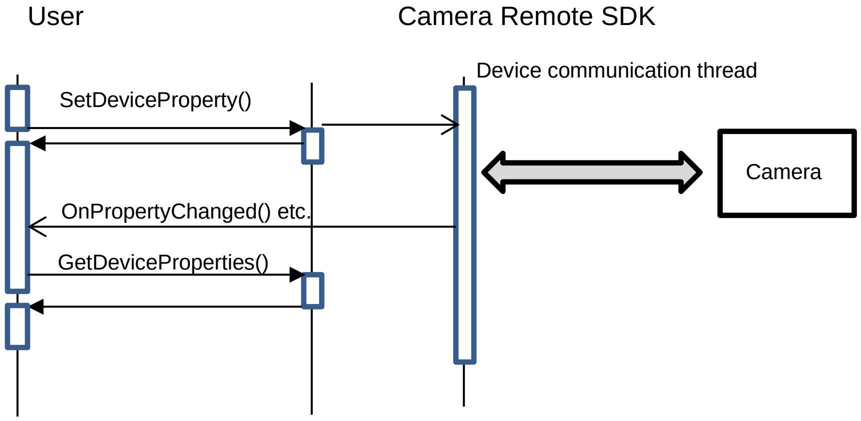
To change camera properties, for example F number, shutter speed, ISO and so on, send change property commands by using SetDeviceProperty(). There are two parameters, the first parameter is the device handle of the target camera, and the second parameter is the CrDeviceProperty class object. In this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

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.

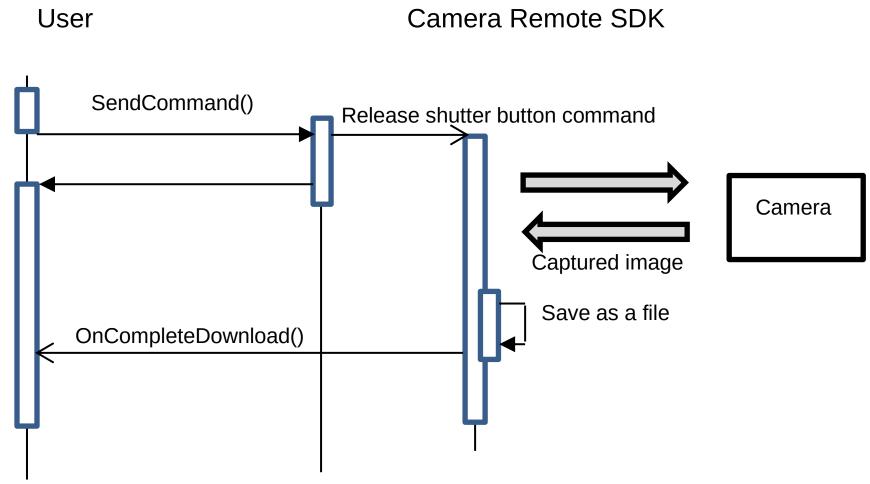
If GetLiveViewImage() returns CrError_Generic_Unknown, it means that there is an issue related to the data communication between the PC and Camera.

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:

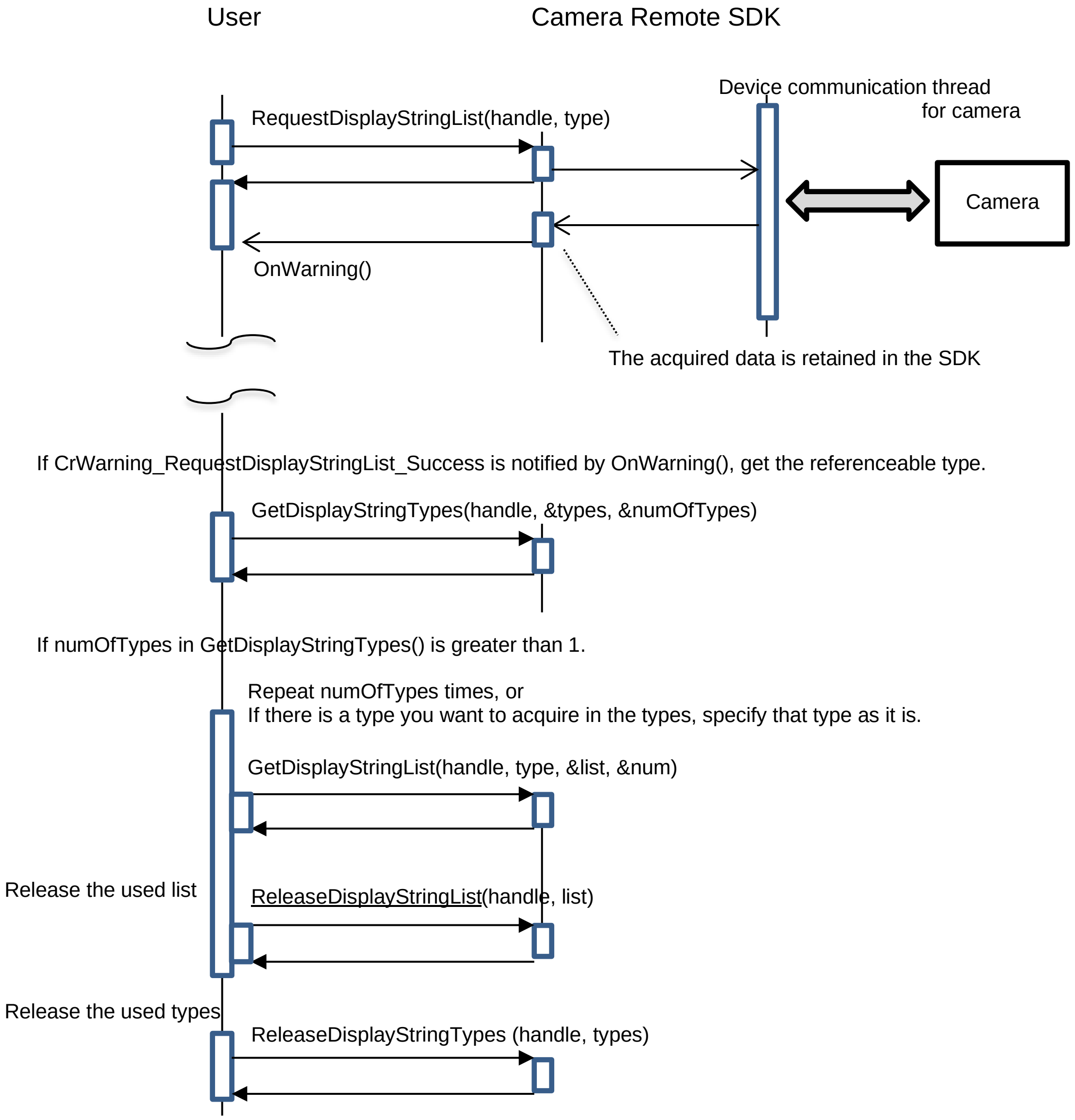
```
SCRSDK::SetSaveInfo(handle, L“C:\Image”, L”ABCDE”, 100);
```

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().


User
Camera Remote SDK
Device communication thread
for camera
RequestDisplayStringList(handle, type)
Camera
OnWarning()
The acquired data is retained in the SDK
If CrWarning_RequestDisplayStringList_Success is notified by OnWarning(), get the referenceable type.
GetDisplayStringTypes(handle, &types, &numOfTypes)
If numOfTypes in GetDisplayStringTypes() is greater than 1.
Repeat numOfTypes times, or
If there is a type you want to acquire in the types, specify that type as it is.
GetDisplayStringList(handle, type, &list, &num)
Release the used list
ReleaseDisplayStringList(handle, list)
Release the used types
ReleaseDisplayStringTypes (handle, types)

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:

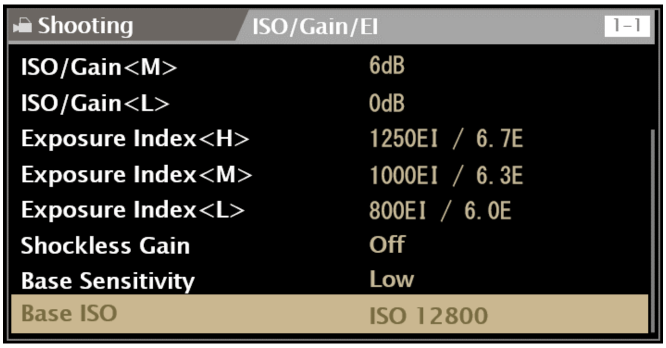
```
                CrInt32u numOfTypes = 0;

                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:

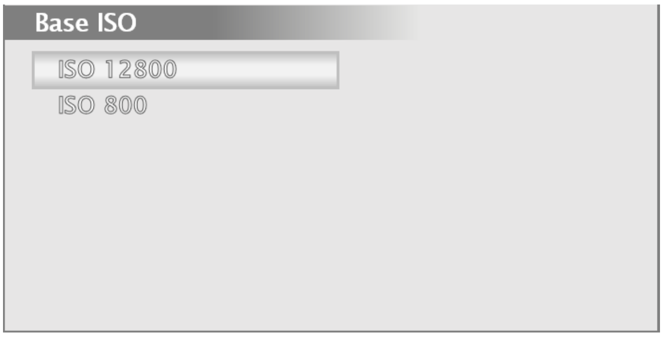
```
for (auto item = m_baseIsoList.begin(); item != m_baseIsoList.end(); ++item) {
                printf("[%d] %s\n", item->first, item->second);
}
```

ex. ILME-FX6 Gain BaseISO Menu



Shooting ISO/Gain/EI 1-1

ISO/Gain<M>	6dB
ISO/Gain<L>	0dB
Exposure Index<H>	1250EI / 6.7E
Exposure Index<M>	1000EI / 6.3E
Exposure Index<L>	800EI / 6.0E
Shockless Gain	Off
Base Sensitivity	Low
Base ISO	ISO 12800



Base ISO

ISO 12800

ISO 800

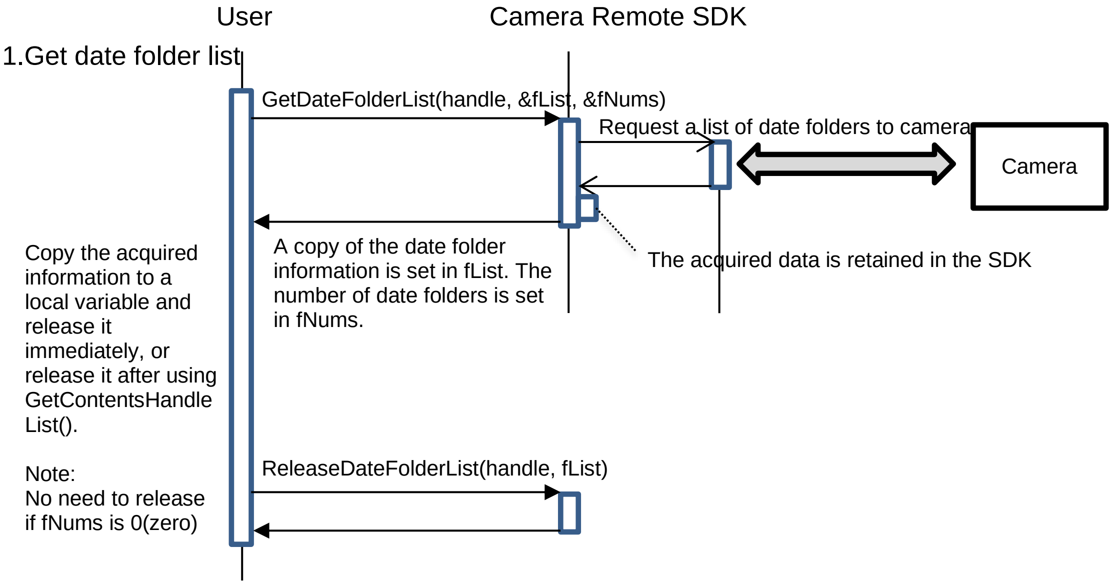
Pull out content stored on media

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

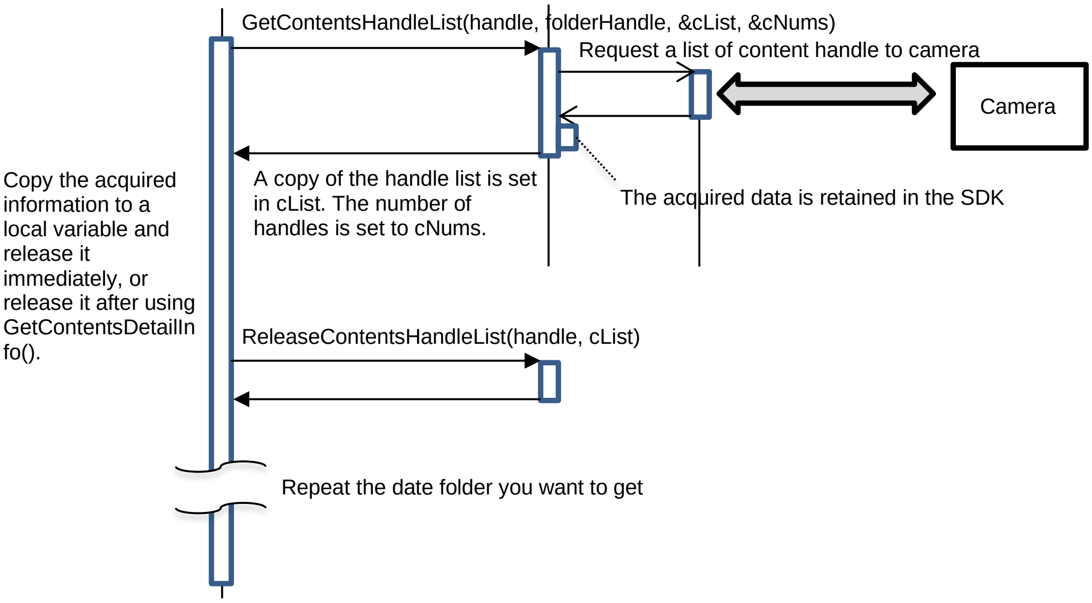
In order to pull content from the media, a content handle is required to identify the content.

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().

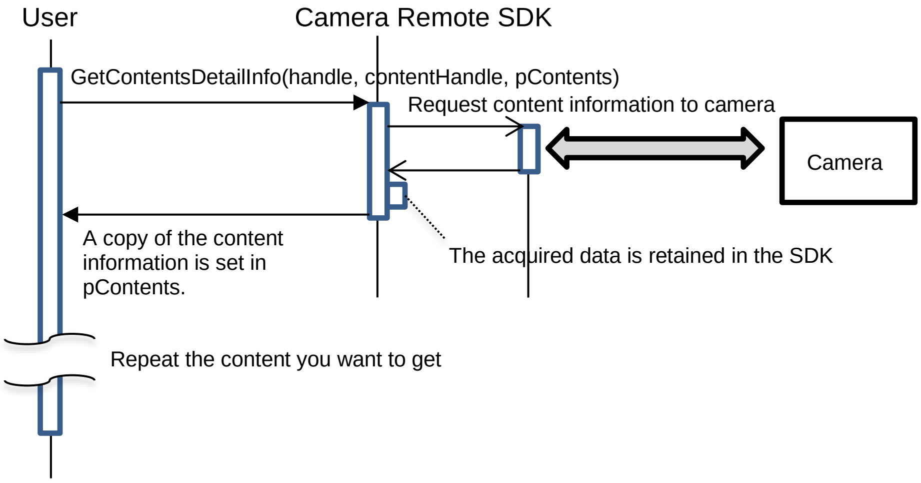
To know the file name and size of the content, get the detailed information with GetContentsDetailInfo().


User
Camera Remote SDK
1.Get date folder list
GetDateFolderList(handle, &fList, &fNums)
Request a list of date folders to camera
Camera
A copy of the date folder information is set in fList. The number of date folders is set in fNums.
The acquired data is retained in the SDK
Copy the acquired information to a local variable and release it immediately, or release it after using GetContentsHandleList().
Note:
No need to release if fNums is 0(zero)
ReleaseDateFolderList(handle, fList)

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


GetContentsHandleList(handle, folderHandle, &cList, &cNums)
Request a list of content handle to camera
Camera
A copy of the handle list is set in cList. The number of handles is set to cNums.
The acquired data is retained in the SDK
Copy the acquired information to a local variable and release it immediately, or release it after using GetContentsDetailInfo().
ReleaseContentsHandleList(handle, cList)
Repeat the date folder you want to get

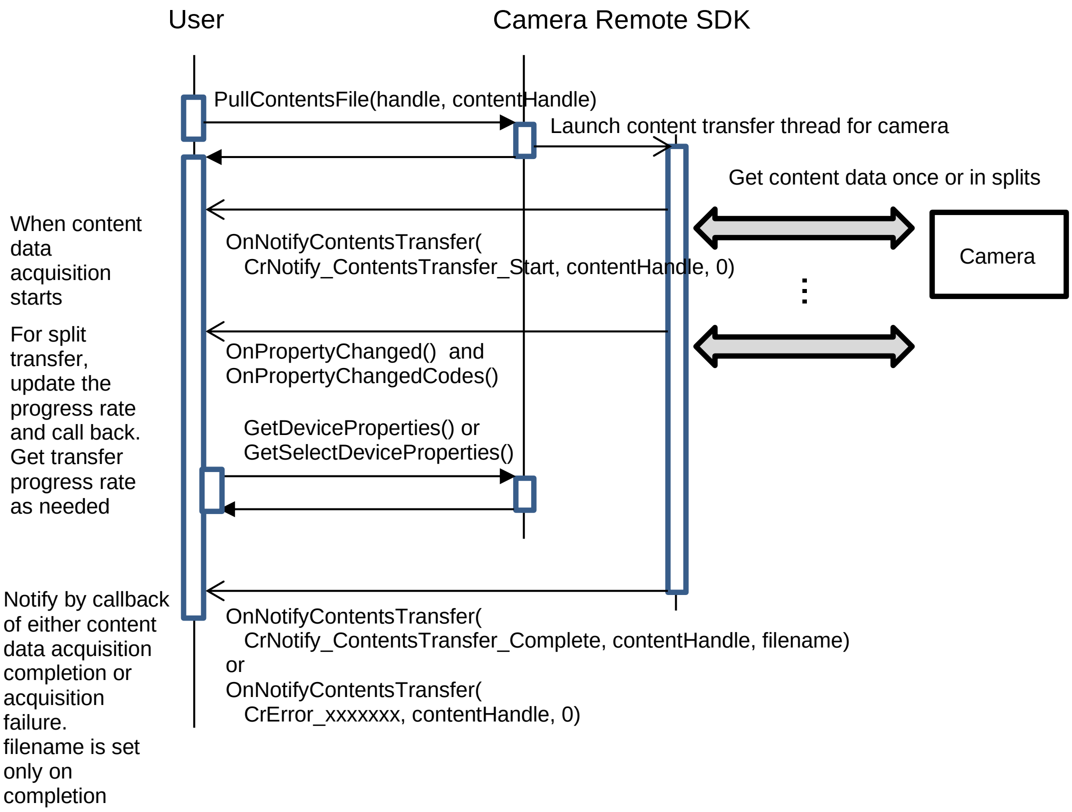
3. Get content detail information


User
Camera Remote SDK
GetContentsDetailInfo(handle, contentHandle, pContents)
Request content information to camera
Camera
A copy of the content information is set in pContents.
The acquired data is retained in the SDK
Repeat the content you want to get

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.


User
Camera Remote SDK
PullContentsFile(handle, contentHandle)
Launch content transfer thread for camera
Get content data once or in splits
Camera
When content data acquisition starts
OnNotifyContentsTransfer(CrNotify_ContentsTransfer_Start, contentHandle, 0)
For split transfer, update the progress rate and call back. Get transfer progress rate as needed
OnPropertyChanged() and OnPropertyChangedCodes()
GetDeviceProperties() or GetSelectDeviceProperties()
Notify by callback of either content data acquisition completion or acquisition failure. filename is set only on completion
OnNotifyContentsTransfer(CrNotify_ContentsTransfer_Complete, contentHandle, filename)
or
OnNotifyContentsTransfer(CrError_xxxxxxx, contentHandle, 0)

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.

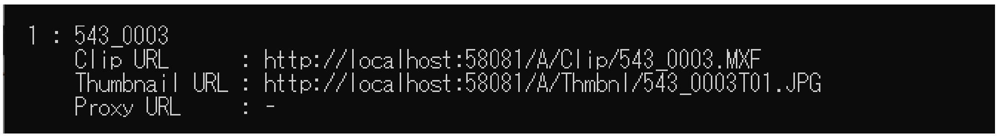
The second parameter specifies the Slot for which you want to get the MediaProfile. The third parameter is a pointer to which the list information of the acquired MediaProfile is written. The fourth parameter is set to the number of acquired MediaProfile and returns.

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);

}
```

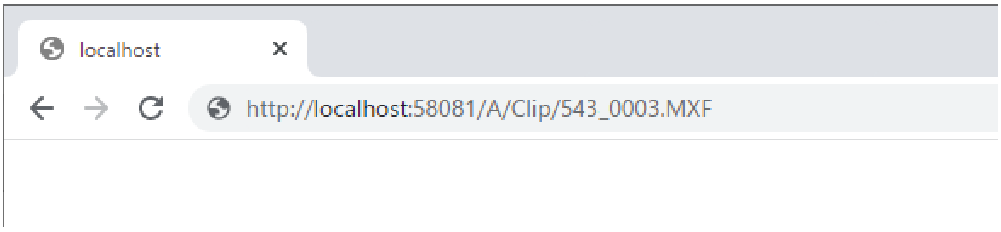
For example, there is "contentUrl" in the information obtained by this API. If you enter those URLs in browser software (Chrome, Safari, etc.) that supports streaming playback, you can play the content.

# Below is an example of Sample Application output



```
1 : 543_0003
    Clip URL      : http://localhost:58081/A/Clip/543_0003.MXF
    Thumbnail URL : http://localhost:58081/A/Thmbnl/543_0003T01.JPG
    Proxy URL     : -
```

Input to web-browser



http://localhost:58081/A/Clip/543_0003.MXF

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, CrInte32u value);

```
Example:

            SCRSDK::SetDeviceSetting(handle, Setting_Key_EnableLiveView, 0);

                         :

            SCRSDK::SetDeviceSetting(handle, Setting_Key_EnableLiveView, 1);
```

The code sample above disables and enables the live view feature. Set Zero to disable and set one to enable the feature.

In this version of Camera Remote SDK, only the Setting_Key_EnableLiveView setting can be set.

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.

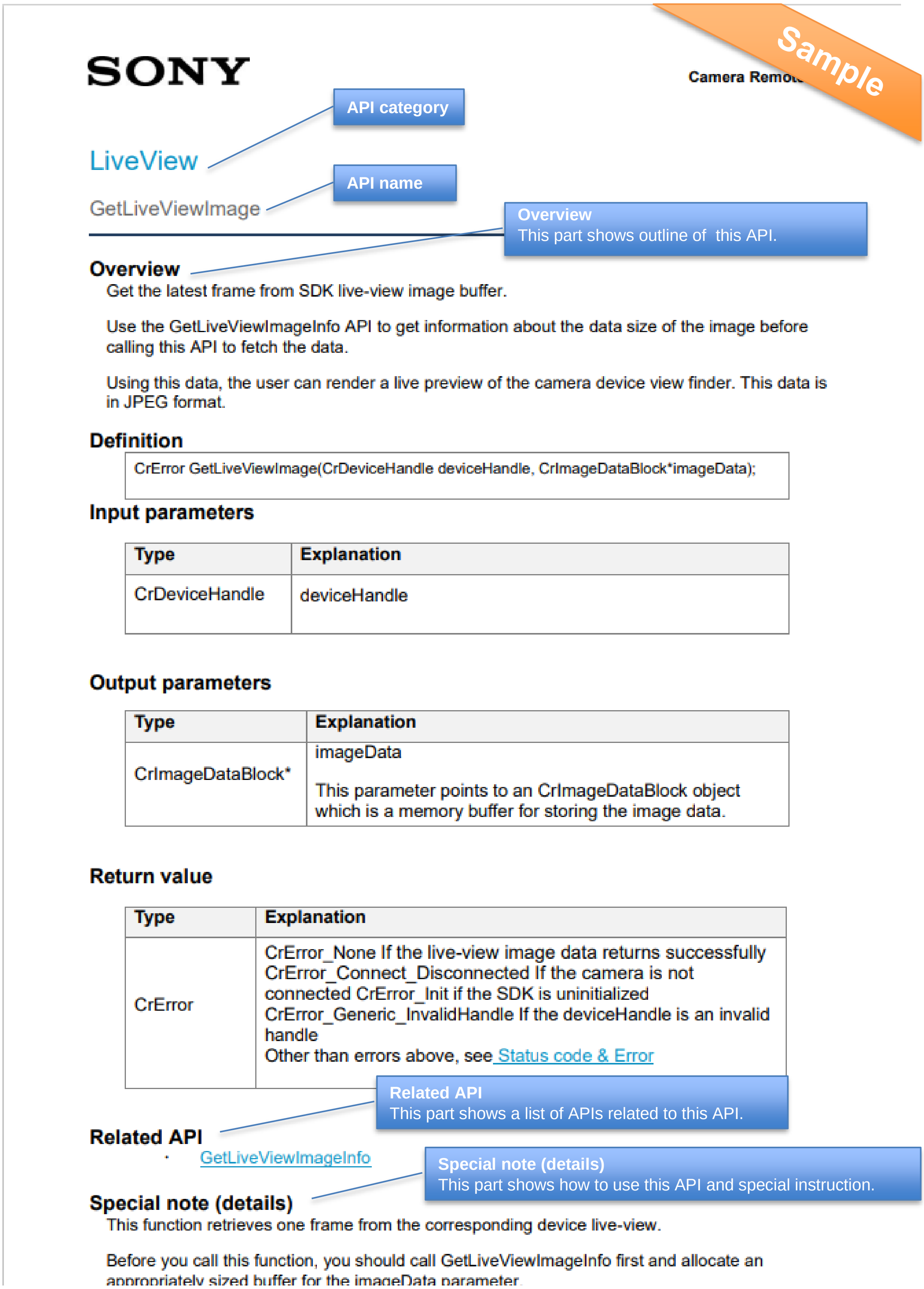
The upload result will be notified by a OnWarning(). If a file for another model or an invalid file is uploaded, CrWarning_CameraSettings_Read_Result_Invalid etc. will be returned.

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.



Sample

Camera Remote

API category

LiveView

API name

GetLiveViewImage

Overview
This part shows outline of this API.

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 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 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 Other than errors above, see Status code & Error

Related API
This part shows a list of APIs related to this API.

Related API

- GetLiveViewImageInfo

Special note (details)
This part shows how to use this API and special instruction.

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.

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 If initialize successfully, the result is true; otherwise, the result is false.

Related API

- Release

Special note (details)

None in particular

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 This parameter is not supported with the current Camera Remote SDK.

Output parameters

Type	Explanation
ICrEnumCameraObjectInfo**	ppEnumCameraObjectInfo This is an input/output parameter. 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. 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 CrError_Init if the SDK is uninitialized CrError_Adaptor_HandlePlugin if any plugin modules are not found 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 Not available.
CrChar*	model Null-terminated device model name string
CrInt16	usbPid Pid for usb devices
CrInt32u	idType For PTP_USB, this is CAMERAOBJECTID_TYPE_USB.
CrInt32u	idSize Size in bytes of the id buffer
CrInt8u*	id A buffer containing device information
CrChar*	connectTypeName A char pointer which points to the null-terminated string of the connection type name of the camera. For PTP_USB, the string is "USB";
CrChar*	adaptorName A char pointer which points to the null-terminated string of the adapter name of the camera. For PTP_USB, the string is "Cr_PTP_USB";
CrChar*	pairingNecessity Call with NULL, because this parameter is not used.

CrInt32u	sshSupport This parameter is optional. 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. 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 Model of the Camera. Use the CrCameraDeviceModelList defined in CrDefines.h.
CrInt8u*	usbSerialNumber Serial number for usb devices. 12byte + Null-terminated refs. To check the USB serial number

Output parameters

Type	Explanation
ICrCameraObjectInfo**	pCameraObjectInfo A pointer to the ICrCameraObjectInfo. Specify the address of a modifiable ICrCameraObjectInfo pointer. Caution: 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 CrError_Api_OutOfModelList CrCameraDeviceModelList If the value does not exist in the model CrError_Api_NotSupportModelOfUSB USB For unsupported model 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 Model of the Camera. Use the CrCameraDeviceModelList defined in CrDefines.h.
CrInt32u	ipAddress IP address of the camera Ex.) 192.168.0.5 = 0x0500A8C0 To convert a dot-separated string notation to a 32-bit value Please set the 1st <-> 7~0bit, the 2nd <-> 15~8bit, the 3rd <-> 23~16bit, and the 4th <-> 31~24bit.
CrInt8u*	macAddress MAC address of the camera. 6byte fixed. 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 This parameter is optional. For SSH authentication models, set CrSSHsupport_ON. Caution: 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 A pointer to the ICrCameraObjectInfo. Specify the address of a modifiable ICrCameraObjectInfo pointer. Notice: 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 CrError_Api_OutOfModelList CrCameraDeviceModelList If the value does not exist in the model CrError_Api_NotSupportModelOfEthernet For unsupported model CrError_Api_InvalidIpAddress If the IP address is judged to be inappropriate 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 he camera which is going to be connected. This parameter is return by ICrEnumCameraObjectInfo::GetCameraObject().

Output parameters

Type	Explanation
char*	fingerprint The fingerprint pointer. Developer prepares a larger buffer to receive fingerprint data, and passes the address of this pointer. When function returns successfully, this parameter will points to a Base64 encoded character. Note: Add the "=" sign for padding. Does not contain Null-terminations.
CrInt32u*	fingerprintSize 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidParameter If the parameter is NULL CrError_Connect_SSH_NotSupported If SSH connection is not supported 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. 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 The camera which is going to be connected. This parameter is return by ICrEnumCameraObjectInfo::GetCameraObject().
IDeviceCallback*	callback 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 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". Note : Switching between RemoteControlMode and ContentsTransferMode cannot be performed while connected. After disconnecting in each mode, reconnect in the desired mode.
CrReconnectingSet	reconnect 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 This parameter is optional. Specify the User name for the SSH authentication. Make it null terminated. For details on how to set the User name for SSH authentication, refer to the help guide for the target camera.
const char*	userPassword This parameter is optional. Specify the password for the SSH authentication. Make it null terminated. For details on how to set the password for SSH authentication, refer to the help guide for the target camera.
const char*	fingerprint This parameter is optional. Specify the fingerprint data obtained by GetFingerprint().
CrInt32u	fingerprintSize This parameter is optional. Specify the length of the fingerprint parameter.

Input/Output parameters

Type	Explanation
CrDeviceHandle*	deviceHandle 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 CrError_Init if the SDK is uninitialized CrError_Generic_Unknown If the pCameraObjectInfo is NULL, and no valid deviceNumber is supplied CrError_Connect_ContentsTransfer_NotSupported Connected to a model that does not support content transfer. Errors starting with CrError_Connect_SSH, such as CrError_Connect_SSH_ServerConnectFailed, indicate an SSH connection error. 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. CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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. CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 The property list pointer. Developers should pass the address of a modifiable CrDeviceProperty pointer. The value of this pointer should be initialized to nullptr. 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 A pointer to an integer which indicates the number of CrDeviceProperty objects in the property list. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 Number of device properties to get.
CrInt32u*	codes List of device property codes to get.

Output parameters

Type	Explanation
CrDeviceProperty**	properties The property list pointer. Developers should pass the address of a modifiable CrDeviceProperty pointer. The value of this pointer should be initialized to nullptr. 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 A pointer to an integer which indicates the number of CrDeviceProperty objects in the property list. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 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. CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 “CrCommnadId_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 This parameter is one of CrCommandId defined in CrCommandData.h.
CrCommandParam	commandParam 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. CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 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 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 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

```
CrInt32u GetLiveViewImageInfo(CrDeviceHandle deviceHandle, CrImageInfo* info);
```

Input parameters

Type	Explanation
CrDeviceHandle	deviceHandle

Output parameters

Type	Explanation
CrImageInfo*	info 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 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 The property list pointer. Developers should pass the address of a modifiable CrLiveViewProperty pointer. The value of this pointer should be initialized to nullptr. The function will make a copy of the SDK-internal CrLiveViewProperty list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of CrLiveViewProperty list. Must be freed with ReleaseLiveViewProperties() after use.
CrInt32*	numOfProperties A pointer to an integer which indicates the number of CrLiveViewProperty objects in the property list. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 Number of live-view properties to get.
CrInt32u*	codes List of live-view property codes to get.

Output parameters

Type	Explanation
CrLiveViewProperty**	properties The property list pointer. Developers should pass the address of a modifiable CrLiveViewProperty pointer. The value of this pointer should be initialized to nullptr. The function will make a copy of the specified CrLiveViewProperty list for the indicated deviceHandle. When function returns successfully, this parameter will point to the copy of CrLiveViewProperty list. Must be freed with ReleaseLiveViewProperties() after use.
CrInt32*	numOfProperties A pointer to an integer which indicates the number of CrLiveViewProperty objects in the property list. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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.

This API can be used as a query to enable or disable status of live-view information for a device.

Definition

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

Input parameters

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

Output parameters

Type	Explanation
CrInt32*	value The current value of the key in question. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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.

This API can be used as a query to enable or disable the live-view information for a device.

Definition

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

Input parameters

Type	Explanation
CrDeviceHandle	deviceHandle
CrInt32u	key Key for the setting to update. In this version of Camera Remote SDK, only the Setting_Key_EnableLiveView setting can be set.
CrInt32u	value The new value for key.

Output parameters

None

Return value

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

Related API

- GetDeviceSetting

Special note (details)

None in particular

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 The local path where images should be saved. This path is also a content transfer path. If you do not specify the filePath parameter of the PullContentsFile function, the path specified by this parameter is used. 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 The prefix to give saved images. This parameter is valid only when shooting in RemoteControlMode. Not used in ContentsTransferMode.
CrInt32	no The starting value to use when enumerating images. 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 CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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. 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. 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. The constant values are in the SDKInfoType enumeration. It is possible to delete camera-specific information with the following values. *SDKINFO_AUTHINFO*

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None on Success 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 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. 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. 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 A pointer to an integer which indicates the number of CrMtpFolderInfo objects in the date folder list. 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 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. CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function. Other than errors above, see Status code & Error

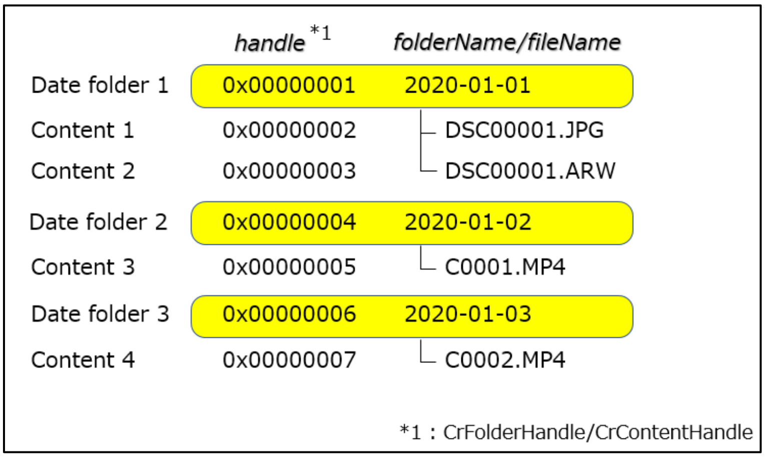
Related API

- 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



	handle *1	folderName/fileName
Date folder 1	0x00000001	2020-01-01
Content 1	0x00000002	├ DSC00001.JPG
Content 2	0x00000003	└ DSC00001.ARW
Date folder 2	0x00000004	2020-01-02
Content 3	0x00000005	└ C0001.MP4
Date folder 3	0x00000006	2020-01-03
Content 4	0x00000007	└ C0002.MP4

*1 : CrFolderHandle/CrContentHandle

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 Specifies one of the date folder handles obtained by the GetDateFolderList function.

Output parameters

Type	Explanation
CrContentHandle **	contentsHandles 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. 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. 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 A pointer to an integer which indicates the number of content in the date folder. 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 CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function. Other than errors above, see Status code & Error

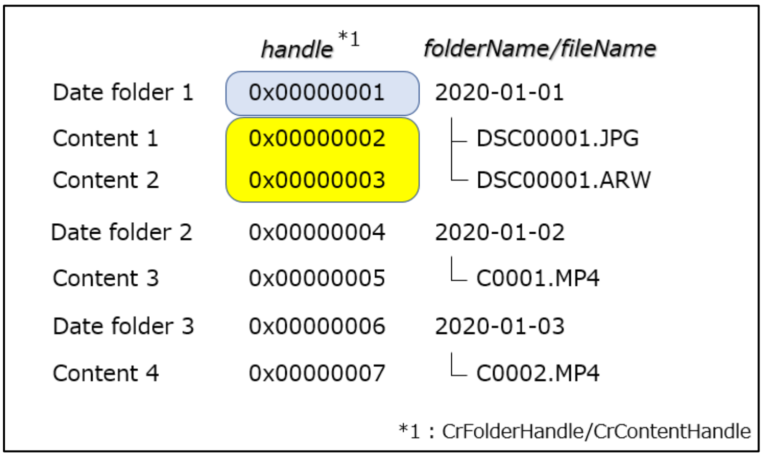
Related API

- 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



	handle *1	*folderName/fileName*
Date folder 1	0x00000001	2020-01-01
Content 1	0x00000002	├ DSC00001.JPG
Content 2	0x00000003	└ DSC00001.ARW
Date folder 2	0x00000004	2020-01-02
Content 3	0x00000005	└ C0001.MP4
Date folder 3	0x00000006	2020-01-03
Content 4	0x00000007	└ C0002.MP4

*1 : CrFolderHandle/CrContentHandle

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 Specifies one of the content handles obtained by the GetContentsHandleList function.

Output parameters

Type	Explanation
CrMtpContentsInfo*	contentsInfo 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. 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. 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 CrError_Contents_InvalidHandle If the content handle specified is invalid CrError_Contents_RejectRequest Returned during the content transfer process. When the content transfer process is completed, re-execute this function. Other than errors above, see Status code & Error

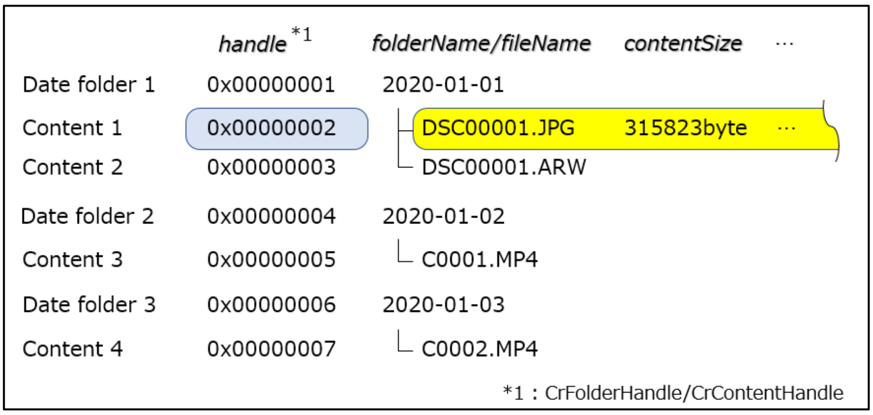
Related API

- 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



	handle *1	*folderName/fileName*	*contentSize*	…
Date folder 1	0x00000001	2020-01-01		
Content 1	0x00000002	├ DSC00001.JPG	315823byte	…
Content 2	0x00000003	└ DSC00001.ARW		
Date folder 2	0x00000004	2020-01-02		
Content 3	0x00000005	└ C0001.MP4		
Date folder 3	0x00000006	2020-01-03		
Content 4	0x00000007	└ C0002.MP4		

*1 : CrFolderHandle/CrContentHandle

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 Date folder list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the date folder list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 Content handle list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the content handle list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 Specifies one of the content handles obtained by the GetContentsHandleList function. Only content whose details have been obtained using the GetContentsDetailInfo function can be specified.
CrPropertyStillImageTransSize	size Specify the size of the still image to be acquired. Specify either CrPropertyStillImageTransSize_Original or CrPropertyStillImageTransSize_SmallSize. When CrPropertyStillImageTransSize_SmallSize is specified 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. 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. If you specify small for the movie, an error is returned.
CrChar*	path This parameter is optional. 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 This parameter is optional. If not specified, the content will be saved with the file name. 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 CrError_Contents_RejectRequest If content cannot be transferred CrError_Generic_NotSupported CrPropertyStillImageTransSize_SmallSize specified for movie content CrError_File_StorageFull Insufficient storage capacity on the host 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 Specifies one of the content handles obtained by the GetContentsHandleList function. Only content whose details have been obtained using the GetContentsDetailInfo function can be specified.

Output parameters

Type	Explanation
CrImageDataBlock*	imageData This parameter points to an CrImageDataBlock object which is a memory buffer for storing the image data. JPEG image data of 160 x 120 pixels is set in the pointer. 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 A type that means the format of a thumbnail image. Developers should pass the address of a modifiable CrFileType variable. Thumbnail images of JPEG content, RAW content, and movie content are in JPEG format. Thumbnail images of HEIF content are in HEIF format. Caution: 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 CrError_Contents_RejectRequest When content is being transferred or thumbnail image data cannot be obtained 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 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.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None on Success 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 The CrDisplayStringType list pointer. Developers should pass the address of a modifiable CrDisplayStringType pointer. The value of this pointer should be initialized to nullptr. 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. Note: It may contain CrDisplayStringType that the SDK does not support.
CrInt32u*	numOfTypes An integer pointer that indicates the number of CrDisplayStringType returned by the types pointer. Developers should pass the address of a modifiable CrInt32u variable.

Return value

Type	Explanation
CrError	CrError_None on Success However, if numOfTypes is zero, even if CrError_None is returned, it should be judged as fail. 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 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. 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 The CrDisplayStringListInfo list pointer. Developers should pass the address of a modifiable CrDisplayStringListInfo pointer. The value of this pointer should be initialized to nullptr. 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. Note: 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 An integer pointer that indicates the number of CrDisplayStringListInfo returned by the list pointer. Developers should pass the address of a modifiable CrInt32u variable.

Return value

Type	Explanation
CrError	CrError_None on Success 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 CrDisplayStringType list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the Type list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 CrDisplayStringListInfo list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 Specifies the type of file to download.
CrChar*	filePath This parameter is optional. 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 This parameter is optional. The extension is fixed to "DAT". If this parameter is not specified, the file will be saved with "Camera model name + _CUMSET.DAT" fixed. 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 CrError_File_StorageFull Insufficient storage capacity on the host. 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 Specifies the type of file to upload.
CrChar*	fileName Path of the file to be uploaded. The extension is fixed to "DAT".

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None on Success 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 Specifies the slot from which to get the MediaProfile. refs. CrMediaProfile

Output parameters

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

Return value

Type	Explanation
CrError	CrError_None on Success 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 CrMediaProfileInfo list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the Type list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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 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 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 An integer pointer that indicates the number of CrLensInformation returned by the list pointer. Developers should pass the address of a modifiable CrInt32u variable.

Return value

Type	Explanation
CrError	CrError_None on Success 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 CrLensInformation list pointer to release.

Output parameters

None

Return value

Type	Explanation
CrError	CrError_None If the Type list is released successfully CrError_Init if the SDK is uninitialized CrError_Generic_InvalidHandle If the deviceHandle is an invalid handle 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

Supported Command

Command supported in the current release.
The Mode column indicate the availability of RemoteControlMode and ContentsTransferMode in “R” and “C”.

Name	Summary	Mode
CrCommandId_Release	Set the shutter button release	R
CrCommandId_MovieRecord	Control Movie Rec button	R
CrCommandId_MediaFormat	Execute Full Format the Media.	R
CrCommandId_MediaQuickFormat	Execute Quick Format the Media.	R
CrCommandId_CancelMediaFormat	Cancel full formatting of media. Full formatting will be stopped midway, but the index data will be cleared and all data will no longer be accessible.	R
CrCommandId_S1andRelease	Shutter Half Release and Release	R
CrCommandId_CancelContentsTransfer	Cancel content transfer.	C
CrCommandId_CameraSettingsReset	Execute Camera Settings Reset	R
CrCommandId_APS_C_or_Full_Switching	Execute APS-C or Full Switching	R
CrCommandId_MovieRecButtonToggle	Execute Movie Rec Button (2nd)	R
CrCommandId_CancelRemoteTouchOperation	Execute Cancel Remote Touch Operation	R

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.

Member Variables

Name	Type	Summary
-	-	-

Member Functions

Signature	Description
Constructor	-
Destructor	-
Copy Constructor	-
void Alloc(const CrInt32u size, const CrInt32u getSetSize)	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. Defined in CrDeviceProperty.h as CrDevicePropertyCode
CrDataType GetValueType()	Get the CrDataType used by this device property. Defined in CrDefines.h as CrDataType.
CrPropertyEnableFlag GetPropertyEnableFlag()	Get the CrPropertyEnableFlag that represents the enable status for this device property. Defined in CrDeviceProperty.h as CrPropertyEnableFlag. When enableFlag is Disable, Setter/Getter API is invalid (not guaranteed)
CrInt64u GetCurrentValue()	Get the current value. 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
CrInt32u GetValueSize()	Get the total number of bytes of settable value set for values pointer.
CrInt8u* GetValues()	Get the pointer of settable values. 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. Defined in CrDeviceProperty.h as CrDevicePropertyCode.
void SetValueType(CrDataType type)	Set the value type to update. Using CrDataType defined by CrDefines.h.
void SetCurrentValue(CrInt64u value)	Set the value to update. If CrDataType Array, only value that exist in the values pointer can be set. 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.

Supported Properties

Properties supported in the current release.
The Mode column indicate the availability of RemoteControlMode and ContentsTransferMode in “R” and “C”.

Name	Summary	Mode
CrDeviceProperty_S1	Get/Set the shutter button half release	R
CrDeviceProperty_AEL	Get the AELock Indication and control AEL button	R
CrDeviceProperty_FEL	Get the FEL Lock Indication and control FEL button	R
CrDeviceProperty_AWBL	Get the AWBLock Indication and control AWBL button	R
CrDeviceProperty_FNumber	Get/Set the Aperture Value（F-Number)	R
CrDeviceProperty_ExposureBiasCompensation	Get/Set the Exposure Bias Compensation	R
CrDeviceProperty_FlashCompensation	Get/Set the Flash Compensation	R
CrDeviceProperty_ShutterSpeed	Get/Set the Shutter Speed	R

CrDeviceProperty_IsoSensitivity	Get/Set the ISO Sensitivity	R
CrDeviceProperty_FocusArea	Get/Set the Focus Area	R
CrDeviceProperty_ExposureProgramMode	Get/Set the Exposure Program Mode	R
CrDeviceProperty_CompressionFileFormatStill	Get/Set the Compression File Format (Still)	R
CrDeviceProperty_FileType	Get/Set the File Format (Still)	R
CrDeviceProperty_MediaSLOT1_FileType	Get/Set the File Format(Still) of media(SLOT1)	R
CrDeviceProperty_MediaSLOT2_FileType	Get/Set the File Format(Still) of media(SLOT2)	R
CrDeviceProperty_JpegQuality	Get/Set the JPEG Quality	R
CrDeviceProperty_MediaSLOT1_JpegQuality	Get/Set the JPEG Quality of media(SLOT1)	R
CrDeviceProperty_MediaSLOT2_JpegQuality	Get/Set the JPEG Quality of media(SLOT2)	R
CrDeviceProperty_WhiteBalance	Get/Set the WhiteBalance	R
CrDeviceProperty_FocusMode	Get/Set the Focus Mode	R
CrDeviceProperty_MeteringMode	Get/Set the Exposure Metering Mode	R
CrDeviceProperty_FlashMode	Get/Set the Flash Mode	R
CrDeviceProperty_WirelessFlash	Get/Set the Wireless Flash Setting	R
CrDeviceProperty_RedEyeReduction	Get/Set the Red Eye Reduction	R
CrDeviceProperty_DriveMode	Get/Set the Drive Mode（Still Capture Mode)	R
CrDeviceProperty_DRO	Get/Set the Dynamic Range Optimizer	R
CrDeviceProperty_ImageSize	Get/Set the Image Size	R
CrDeviceProperty_MediaSLOT1_ImageSize	Get/Set the Image Size of media(SLOT1)	R
CrDeviceProperty_MediaSLOT2_ImageSize	Get/Set the Image Size of media(SLOT2)	R
CrDeviceProperty_AspectRatio	Get/Set the Aspect Ratio	R
CrDeviceProperty_PictureEffect	Get/Set the Picture Effect Value	R
CrDeviceProperty_Colortemp	Get/Set the Color Temperature	R
CrDeviceProperty_ColorTuningAB	Get/Set the Biaxial Fine Tuning A-B Direction	R
CrDeviceProperty_ColorTuningGM	Get/Set the Biaxial Fine Tuning G-M Direction	R
CrDeviceProperty_LiveViewDisplayEffect	Get/Set the Live View Display Effect	R
CrDeviceProperty_StillImageStoreDestination	Get the information of Still Image Save Destination	R

CrDeviceProperty_PriorityKeySettings	Get/Set the Position Key Setting	R
CrDeviceProperty_Focus_Magnifier_Setting	Get/Set the Focus Magnifier Setting	R
CrDeviceProperty_DateTime_Settings	Set the Date and Time	R
CrDeviceProperty_NearFar	Get/Set the Near/Far	R
CrDeviceProperty_AF_Area_Position	Execute set AF Area Position(x,y)	R
CrDeviceProperty_Zoom_Scale	Get/Set the Zoom Scale	R
CrDeviceProperty_Zoom_Setting	Get/Set the Zoom Setting	R
CrDeviceProperty_Zoom_Operation	Execute the Zoom Operation	R
CrDeviceProperty_Movie_File_Format	Get/Set the File Format(Movie)	R
CrDeviceProperty_Movie_Recording_Setting	Get/Set the Recording Setting(Movie)	R
CrDeviceProperty_Movie_Recording_FrameRateSetting	Get/Set the Recording Frame Rate Setting (Movie)	R
CrDeviceProperty_Interval_Rec_Mode	Get the Interval REC Mode	R
CrDeviceProperty_Still_Image_Trans_Size	Get/Set the Still Image Trans Size	R
CrDeviceProperty_RAW_J_PC_Save_Image	Get/Set the RAW+J PC Save Image	R
CrDeviceProperty_LiveView_Image_Quality	Get/Set the LiveView Quality	R
CrDeviceProperty_CustomWB_Capture_Standby	Get/Set the Custom WB Capture Standby	R
CrDeviceProperty_CustomWB_Capture_Standby_Cancel	Get/Set the Custom WB Capture Standby Cancel	R
CrDeviceProperty_CustomWB_Capture	Execute the Custom WB Capture	R
CrDeviceProperty_SnapshotInfo	Get the Shooting File Info	R
CrDeviceProperty_BatteryRemain	Get the Battery Remaining (%)	R/C
CrDeviceProperty_BatteryLevel	Get the Battery Level Indicator	R/C
CrDeviceProperty_RecordingState	Get the Movie Recording State	R
CrDeviceProperty_LiveViewStatus	LiveView Status	R
CrDeviceProperty_FocusIndication	Get the Focus Indication	R
CrDeviceProperty_MediaSLOT1_Status	Get the Media (SLOT1) Status	R
CrDeviceProperty_MediaSLOT1_RemainingNumber	Get the Remaining number shots of Media (SLOT1)	R
CrDeviceProperty_MediaSLOT1_RemainingTime	Get the Remaining shooting time of Media (SLOT1)	R
CrDeviceProperty_MediaSLOT1_FormatEnableStatus	Get the Media Format Enable Status(SLOT1)	R

CrDeviceProperty_MediaSLOT1_QuickFormatEnableStatus	Get the Media Quick Format Enable Status(SLOT1)	R
CrDeviceProperty_MediaSLOT2_Status	Get the Media (SLOT2) Status	R
CrDeviceProperty_MediaSLOT2_RemainingNumber	Get the Remaining number shots of Media (SLOT2)	R
CrDeviceProperty_MediaSLOT2_RemainingTime	Get the Remaining shooting time of Media (SLOT2)	R
CrDeviceProperty_MediaSLOT2_FormatEnableStatus	Get the Media Format Enable Status(SLOT2)	R
CrDeviceProperty_MediaSLOT2_QuickFormatEnableStatus	Get the Media Quick Format Enable Status(SLOT2)	R
CrDeviceProperty_Media_FormatProgressRate	Get the Media Format Progress Rate	R
CrDeviceProperty_Interval_Rec_Status	Get the Interval REC Status	R
CrDeviceProperty_CustomWB_Execution_State	Get the Custom WB Execution State	R
CrDeviceProperty_CustomWB_Capturable_Area	Get the Custom WB Capturable Area(x,y)	R
CrDeviceProperty_CustomWB_Capture_Frame_Size	Get the Custom WB Capture Frame Size(x,y)	R
CrDeviceProperty_CustomWB_Capture_Operation	Get the Custom WB Capture Operation Enable Status	R
CrDeviceProperty_Zoom_Operation_Status	Get the Zoom Operation Enable Status	R
CrDeviceProperty_Zoom_Bar_Information	Get the Zoom Bar Information	R
CrDeviceProperty_Zoom_Type_Status	Get the Zoom Type Status	R
CrDeviceProperty_RAW_FileCompressionType	Get/Set the compression type of RAW file	R
CrDeviceProperty_MediaSLOT1_RAW_FileCompressionType	Get/Set the compression type of RAW file in media(SLOT1)	R
CrDeviceProperty_MediaSLOT2_RAW_FileCompressionType	Get/Set the compression type of RAW file in media(SLOT2)	R
CrDeviceProperty_Cancel_Media_FormatEnableStatus	Get whether the media format is cancelable	R
CrDeviceProperty_ZoomAndFocusPosition_Save	Get/Set the Save Zoom&FocusPosition Preset	R
CrDeviceProperty_ZoomAndFocusPosition_Load	Get/Set the Load Zoom&FocusPosition Preset	R
CrDeviceProperty_Remocon_Zoom_Speed_Type	Get/Set the Remocon Zoom Speed Type	R
CrDeviceProperty_Zoom_Speed_Range	Get the Zoom Speed Range	R

CrDeviceProperty_SdkControlMode	Get the Sdk Control Mode	R/C
CrDeviceProperty_ContentsTransferStatus	Get the content transfer status	C
CrDeviceProperty_ContentsTransferCancelEnableStatus	Get the cancelability status of content transfer.	C
CrDeviceProperty_ContentsTransferProgress	Gets the handle and progress of the content during transfer	C
CrDeviceProperty_IrisModeSetting	Get/Set the Iris Mode Setting	R
CrDeviceProperty_ShutterModeSetting	Get/Set the Shutter Mode Setting	R
CrDeviceProperty_GainControlSetting	Get/Set the Gain Control Setting	R
CrDeviceProperty_GainBaseIsoSensitivity	Get/Set the Gain Base Iso Sensitivity	R
CrDeviceProperty_GainBaseSensitivity	Get/Set the Gain Base Sensitivity	R
CrDeviceProperty_ExposureIndex	Get/Set the Exposure Index	R
CrDeviceProperty_BaseLookValue	Get/Set the BaseLook Value	R
CrDeviceProperty_PlaybackMedia	Get/Set the Playback Media	R
CrDeviceProperty_DispModeCandidate	Get the Monitor DISP(Screen Display) Mode Candidate	R
CrDeviceProperty_DispModeSetting	Get/Set the Monitor DISP(Screen Display) Mode Setting	R
CrDeviceProperty_DispMode	Get/Set the Monitor DISP(Screen Display) Mode	R
CrDeviceProperty_TouchOperation	Get/Set the Touch Operation	R
CrDeviceProperty_SelectFinder	Get/Set the Select Finder/Monitor	R
CrDeviceProperty_AutoPowerOffTemperature	Get/Set the Auto Power OFF Temperature	R
CrDeviceProperty_BodyKeyLock	Get/Set the Body Key Lock	R
CrDeviceProperty_ImageID_Num_Setting	Get/Set the Image ID(Numerical Value) Setting	R
CrDeviceProperty_ImageID_Num	Get/Set the Image ID(Numerical Value)	R
CrDeviceProperty_ImageID_String	Get/Set the Image ID(String)	R
CrDeviceProperty_ExposureCtrlType	Get/Set the Exposure Control Mode	R
CrDeviceProperty_MonitorLUTSetting	Get/Set the Monitor LUT Setting(All Line)	R
CrDeviceProperty_IsoCurrentSensitivity	Get the ISO Current Sensitivity	R
CrDeviceProperty_CameraSetting_SaveOperationEnableStatus	Get the Camera-Setting Save Operation	R
CrDeviceProperty_CameraSetting_ReadOperationEnableStatus	Get the Camera-Setting Read Operation	R

CrDeviceProperty_CameraSetting_SaveRead_State	Get the Camera-Setting Save/Read State	R
CrDeviceProperty_CameraSettingsResetEnableStatus	Get the Camera Setting Reset Enable Status	R
CrDeviceProperty_APS_C_or_Full_SwitchingSetting	Get the APS-C or Full Switching Setting	R
CrDeviceProperty_APS_C_or_Full_SwitchingEnableStatus	Get the APS-C or Full Switching Enable Status	R
CrDeviceProperty_FocalDistanceInMeter	Get/Set the Focal Distance in Meter	R
CrDeviceProperty_FocalDistanceInFeet	Get/Set the Focal Distance in Feet	R
CrDeviceProperty_FocalDistanceUnitSetting	Get/Set the Focal Distance Unit Setting	R
CrDeviceProperty_DigitalZoomScale	Get/Set the Digital Zoom Scale	R
CrDeviceProperty_ZoomDistance	Get/Set the Zoom Distance	R
CrDeviceProperty_ZoomDistanceUnitSetting	Get/Set the Zoom Distance Unit Setting	R
CrDeviceProperty_ShutterModeStatus	Get/Set the Shutter Mode Status	R
CrDeviceProperty_ShutterSlow	Get/Set the Shutter Slow	R
CrDeviceProperty_ShutterSlowFrames	Get/Set the Shutter Slow Frames	R
CrDeviceProperty_Movie_Recording_ResolutionForMain	Get/Set the Recording Resolution For Main(Movie)	R
CrDeviceProperty_Movie_Recording_ResolutionForProxy	Get/Set the Recording Resolution For Proxy(Movie)	R
CrDeviceProperty_Movie_Recording_FrameRateProxySetting	Get/Set the Recording Frame Rate Proxy Setting(Movie)	R
CrDeviceProperty_MovieShootingMode	Get/Set the Movie Shooting Mode	R
CrDeviceProperty_MovieShootingModeColorGamut	Get/Set the Movie Shooting Mode Color Gamut	R
CrDeviceProperty_MovieShootingModeTargetDisplay	Get/Set the Movie Shooting Mode Target Display	R
CrDeviceProperty_DepthOfFieldAdjustmentMode	Get/Set the Depth of Field Adjustment Mode	R
CrDeviceProperty_DepthOfFieldAdjustmentInterlockingMode	Get the Depth of Field Adjustment Interlocking Mode State	R
CrDeviceProperty_ColortempStep	Set the Color Temperature	R
CrDeviceProperty_WhiteBalanceModeSetting	Get/Set the White Balance Mode Setting	R
CrDeviceProperty_WhiteBalanceTint	Get/Set the White Balance Tint	R
CrDeviceProperty_WhiteBalanceTintStep	Set the White Balance Tint	R
CrDeviceProperty_Focus_Operation	Execute the Focus Operation	R

CrDeviceProperty_Focus_Speed_Range	Get the Focus Speed Range	R
CrDeviceProperty_ShutterECSSetting	Get/Set the Shutter ECS Setting	R
CrDeviceProperty_ShutterECSNumber	Get/Set the Shutter ECS Number	R
CrDeviceProperty_ShutterECSNumberStep	Set the Shutter ECS Number Step	R
CrDeviceProperty_ShutterECSFrequency	Get/Set the Shutter ECS Frequency	R
CrDeviceProperty_ButtonAssignmentAssignable1	Get/Set the Button Assignment Assignable 1	R
CrDeviceProperty_ButtonAssignmentAssignable2	Get/Set the Button Assignment Assignable 2	R
CrDeviceProperty_ButtonAssignmentAssignable3	Get/Set the Button Assignment Assignable 3	R
CrDeviceProperty_ButtonAssignmentAssignable4	Get/Set the Button Assignment Assignable 4	R
CrDeviceProperty_ButtonAssignmentAssignable5	Get/Set the Button Assignment Assignable 5	R
CrDeviceProperty_ButtonAssignmentAssignable6	Get/Set the Button Assignment Assignable 6	R
CrDeviceProperty_ButtonAssignmentAssignable7	Get/Set the Button Assignment Assignable 7	R
CrDeviceProperty_ButtonAssignmentAssignable8	Get/Set the Button Assignment Assignable 8	R
CrDeviceProperty_ButtonAssignmentAssignable9	Get/Set the Button Assignment Assignable 9	R
CrDeviceProperty_ButtonAssignmentLensAssignable1	Get/Set the Button Assignment LensAssignable 1	R
CrDeviceProperty_AssignableButton1	Get/Set the Assignable Button 1	R
CrDeviceProperty_AssignableButton2	Get/Set the Assignable Button 2	R
CrDeviceProperty_AssignableButton3	Get/Set the Assignable Button 3	R
CrDeviceProperty_AssignableButton4	Get/Set the Assignable Button 4	R
CrDeviceProperty_AssignableButton5	Get/Set the Assignable Button 5	R
CrDeviceProperty_AssignableButton6	Get/Set the Assignable Button 6	R
CrDeviceProperty_AssignableButton7	Get/Set the Assignable Button 7	R
CrDeviceProperty_AssignableButton8	Get/Set the Assignable Button 8	R
CrDeviceProperty_AssignableButton9	Get/Set the Assignable Button 9	R
CrDeviceProperty_LensAssignableButton1	Get/Set the LensAssignable Button 1	R
CrDeviceProperty_FocusModeSetting	Get/Set the Focus Mode Setting	R
CrDeviceProperty_ShutterAngle	Get/Set the Shutter Angle	R
CrDeviceProperty_ShutterSetting	Get/Set the Shutter Setting	R
CrDeviceProperty_ShutterMode	Get/Set the Shutter Mode	R

CrDeviceProperty_ShutterSpeedValue	Get/Set the Shutter Speed Value	R
CrDeviceProperty_ShutterSpeedCurrentValue	Get the Shutter Speed Current Value	R
CrDeviceProperty_NDFilter	Get/Set the ND Filter	R
CrDeviceProperty_NDFilterMode	Get the ND Filter Mode	R
CrDeviceProperty_NDFilterModeSetting	Get/Set the ND Filter Mode Setting	R
CrDeviceProperty_NDFilterValue	Get/Set the ND Filter Value	R
CrDeviceProperty_GainUnitSetting	Get/Set the Gain Unit Setting	R
CrDeviceProperty_GaindBValue	Get/Set the Gain dB Value	R
CrDeviceProperty_GaindBCurrentValue	Get the Gain dB Current Value	R
CrDeviceProperty_AWB	Get/Set the AWB	R
CrDeviceProperty_SceneFileIndex	Get/Set the SceneFile Index	R
CrDeviceProperty_CurrentSceneFileEdited	Get the Current SceneFile Edited Info	R
CrDeviceProperty_MoviePlayButton	Get/Set the Movie Play button	R
CrDeviceProperty_MoviePlayPauseButton	Get/Set the Movie Play Pause button	R
CrDeviceProperty_MoviePlayStopButton	Get/Set the Movie Play Stop button	R
CrDeviceProperty_MovieForwardButton	Get/Set the Movie Forward button	R
CrDeviceProperty_MovieRewindButton	Get/Set the Movie Rewind button	R
CrDeviceProperty_MovieNextButton	Get/Set the Movie Next button	R
CrDeviceProperty_MoviePrevButton	Get/Set the Movie Prev button	R
CrDeviceProperty_MovieRecReviewButton	Get/Set the Movie RecReview button	R
CrDeviceProperty_FaceEyeDetectionAF	Get/Set Face Eye Detection AF	R
CrDeviceProperty_AFTransitionSpeed	Get/Set AF Transition Speed	R
CrDeviceProperty_AFSubjShiftSens	Get/Set AF Subj Shift Sens	R
CrDeviceProperty_AFAssist	Get/Set the AF Assist	R
CrDeviceProperty_NDPresetOrVariableSwitchingSetting	Get/Set the ND PRESET or VARIABLE Switching Setting	R
CrDeviceProperty_FunctionOfRemoteTouchOperation	Get/Set the Function of Remote Touch Operation	R
CrDeviceProperty_RemoteTouchOperation	Execute Remote Touch Operation(x,y)	R
CrDeviceProperty_MoviePlayingState	Get the Movie Playing State	R
CrDeviceProperty_MoviePlayingSpeed	Get Movie Playing Speed	R
CrDeviceProperty_MediaSLOT1Player	Get the Media SLOT1 Player	R
CrDeviceProperty_MediaSLOT2Player	Get the Media SLOT2 Player	R
CrDeviceProperty_BatteryRemainDisplayUnit	Get/Set the Battery Remain Display Unit	R

CrDeviceProperty_BatteryRemainingInMinutes	Get the Battery Remaining in minutes	R
CrDeviceProperty_BatteryRemainingInVoltage	Get the Battery Remaining in voltage	R
CrDeviceProperty_PowerSource	Get/Set the Power Source	R
CrDeviceProperty_DCVoltage	Get the DC voltage	R
CrDeviceProperty_FocusTouchSpotStatus	Get the Focus TouchSpot Status	R
CrDeviceProperty_FocusTrackingStatus	Get the Focus Tracking Status	R
CrDeviceProperty_RecorderClipName	Get Recorder Clip Name Create by The Next Rec.	R
CrDeviceProperty_RecorderControlMainSetting	Get the Recorder Control Main Setting	R
CrDeviceProperty_RecorderControlProxySetting	Get/Set the Recorder Control Proxy Setting	R
CrDeviceProperty_RecorderStartMain	Get the Recorder Start Main	R
CrDeviceProperty_RecorderStartProxy	Get the Recorder Start Proxy	R
CrDeviceProperty_RecorderMainStatus	Get the Recorder Main Status	R
CrDeviceProperty_RecorderProxyStatus	Get the Recorder Proxy Status	R
CrDeviceProperty_RecorderExtRawStatus	Get the Recorder Ext Raw Status	R
CrDeviceProperty_RecorderSaveDestination	Get the information of Recorder Save Destination	R
CrDeviceProperty_AssignableButtonIndicator1	Get the Assignable Button Indicator 1	R
CrDeviceProperty_AssignableButtonIndicator2	Get the Assignable Button Indicator 2	R
CrDeviceProperty_AssignableButtonIndicator3	Get the Assignable Button Indicator 3	R
CrDeviceProperty_AssignableButtonIndicator4	Get the Assignable Button Indicator 4	R
CrDeviceProperty_AssignableButtonIndicator5	Get the Assignable Button Indicator 5	R
CrDeviceProperty_AssignableButtonIndicator6	Get the Assignable Button Indicator 6	R
CrDeviceProperty_AssignableButtonIndicator7	Get the Assignable Button Indicator 7	R
CrDeviceProperty_AssignableButtonIndicator8	Get the Assignable Button Indicator 8	R
CrDeviceProperty_AssignableButtonIndicator9	Get the Assignable Button Indicator 9	R
CrDeviceProperty_LensAssignableButtonIndicator1	Get the LensAssignable Button Indicator 1	R
CrDeviceProperty_SoftwareVersion	Get the Software Version	R

CrDeviceProperty_MovieRecButtonToggleEnableStatus	Get the Movie Rec Button (Toggle) Enable Status	R
CrDeviceProperty_RemoteTouchOperationEnableStatus	Get the Remote Touch Operation Enable Status	R
CrDeviceProperty_CancelRemoteTouchOperationEnableStatus	Get the Cancel Remote Touch Operation Enable Status	R
CrDeviceProperty_LensInformationEnableStatus	Get the Lens Information Enable Status	R
CrDeviceProperty_FollowFocusPositionSetting	Get/Set the Follow Focus Position	R
CrDeviceProperty_FollowFocusPositionCurrentValue	Get the Follow Focus Position Current Value	R
CrDeviceProperty_FocusBracketShotNumber	Get/Set the Focus Bracket Shot Num	R
CrDeviceProperty_FocusBracketFocusRange	Get/Set the Focus Bracket Focus Range	R
CrDeviceProperty_FocusBracketShootingStatus	Get the Focus Bracket Shooting Status	R

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. 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. This pointer is set to CrFocusFrameInfo or CrMagPosInfo.

CrFocusFrameInfo

Used to retrieve live-view frame info.

Member Variables

Name	Type	Summary
type	CrFocusFrameType	The type of focus used Defined in CrDeviceProperty.h as CrFocusFrameType
state	CrFocusFrameState	The state of the focus frame Defined in CrDeviceProperty.h as CrFocusFrameState
priority	CrInt8u	-
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	-

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	-

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. format : "YYYYMMDDThhmmss" 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. Note : The AVCHD file name is in "YYYYMMDDhhmmss" format (datetime). 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
CrDisplayStringType_AllList	Request all list types
CrDisplayStringType_BaseLook_AELevelOffset_ExposureValue	Rightmost column of Menu > Paint/Look > Base Look > Select ILC is not supported.
CrDisplayStringType_BaseLook_Input_Display	Input column of Menu > Paint/Look > Base Look > Select ILC is not supported.
CrDisplayStringType_BaseLook_Name_Display	ILC: Table of Menu > Exposure/Color > Color/Tone > Select LUT Non ILC: Base Look Name column of Menu > Paint/Look > Base Look > Select
CrDisplayStringType_BaseLook_Output_Display	Output column of Menu > Paint/Look > Base Look > Select ILC is not supported.
CrDisplayStringType_SceneFile_Name_Display	Base Look Name column of Menu > Paint/Look > Scene File > Recall when Menu > Project > Base Setting > Shooting Mode is “Custom” ILC is not supported.
CrDisplayStringType_ShootingMode_Cinema_ColorGamut_Display	Menu > Project > Cine EI Setting > Color Gamut ILC is not supported.
CrDisplayStringType_ShootingMode_TargetDisplay_Display	Menu > Project > Base Setting > Target Display ILC is not supported.
CrDisplayStringType_Camera_Gain_BaseISO_Display	ILC: Menu > Exposure/Color > Exposure > Base ISO Non ILC: Menu > Shooting > ISO/Gain/EI > BaseISO

CrDisplayStringType_Video_EIGain_Display	ILC: Menu > Exposure/Color > Exposure > Exposure Index Non ILC: Menu > Shooting > ISO/Gain/EI > Exposure Index<H>, <M>, <L> when Menu > Project > Base Setting > Shooting Mode is “Cine EI”
CrDisplayStringType_Button_Assign_Display	Menu > Project > Assignable Button ILC is not supported.
CrDisplayStringType_Button_Assign_ShortDisplay	Abbreviation string for Menu > Project > Assignable Button. This menu is not in the camera body. ILC is not supported.

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 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. When capture image transfer is complete, the type parameter is 0xFFFFFFFF. 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. 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. The filename parameter is the name (including path) of the content that will be set when the content transfer is complete. 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 (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(CrInt32u error) 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_PortfowardFailed	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_InvaildProperty	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_EnumDecvice	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 Exceeded the number of connectable devices
CrWarning_File_StorageFull	Warning: host storage is almost full 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: DataTime 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: Menu > Project > Base Setting > Shooting Mode, or Target Display update. ILC is not supported.
CrWarning_DisplayListChanged_BaseLook_InputDisplayList	Same of CrWarning_DisplayListChanged_BaseLook_AELevelOffsetExposureValueList ILC is not supported.

CrWarning_DisplayListChanged_BaseLook_NameDisplayList	ILC Warning: LUT registration, deletion, and editing Non ILC Warning : Same of CrWarning_DisplayListChanged_BaseLook_AELevelOffsetExposureValueList
CrWarning_DisplayListChanged_BaseLook_OutputDisplayList	Same of CrWarning_DisplayListChanged_BaseLook_AELevelOffsetExposureValueList ILC is not supported.
CrWarning_DisplayListChanged_SceneFile_NameDisplayList	Warning: Scene File save at Menu > Paint/Look > Scene File > Store, or Menu > Project > Base Setting > Target Display update. ILC is not supported.
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	ILC Warning: Menu > Shooting > Image Quality/Rec > Log Shooting Setting > Log Shooting update Non ILC Warning: Menu > Shooting > ISO/Gain/EI > Base ISO update
CrWarning_DisplayListChanged_Button_AssignDisplayList	Warning: Button assign display list update
CrWarning_DisplayListChanged_Button_AssignShortDisplayList	Warning: Button assign short display list update
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)

CrNotify

Name	Summary
CrNotify_All_Download_Complete	Notification: download completed
CrNotify_Captured_Event	Notification: Still image capture complete. 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 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 Note: After starting movie recording, please check the movie recording status with CrDeviceProperty_RecordingState. Caution: The below models can be start or stop with the “Down”, but please execute “Up” after “Down” at once. 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 Ex. “CrCommandId_MediaFormat” with “CrCommandParam_Up”
CrCommandParam_Down	Specify when initializing the media in SLOT2 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 Ex. “CrCommandId_MediaQuickFormat” with “CrCommandParam_Up”
CrCommandParam_Down	Specify when quick and simple initializing the media in SLOT2 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. After executing Down, please release the Down state by executing Up. When CrDeviceProperty_Cancel_Media_FormatEnableStatus is Enable, it is possible to cancel Full format(CrCommandId_MediaFormat) by sending this command. 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 After executing “Down”, send “Up” to cancel the Down status.

CrCommandId_CancelContentsTransfer

Cancel content transfer

Parameter Code	Explanation
CrCommandParam_Down	Specify when canceling the content transfer process 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. Valid when CrDeviceProperty_CameraSettingsResetEnableStatus is Enable. This operation resets the camera settings and restarts the camera. Caution: 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. 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. Valid when CrDeviceProperty_MovieRecButtonToggleEnableStatus is Enable. Note: After starting movie recording, please check the movie recording status with CrDeviceProperty_RecordingState.

CrCommandId_CancelRemoteTouchOperation

Cancel Remote Touch Operation

Parameter Code	Explanation
CrCommandParam_Down	Specify when canceling the Remote Touch Operation Check the CrDeviceProperty_CancelRemoteTouchOperationEnableStatus status to see if you can cancel or not.

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. e.g.) 0x0190 =400 (means F-4) 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. e.g.) 0xEC78 = -5000 (means -5.0Ev) 0x0000 = 0 (means 0.0Ev) 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. e.g.) 0xEC78 = -5000 (means -5.0Ev) 0x0000 = 0 (means 0.0Ev) 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) In the case of the shutter speed is displayed as "Real Number" on the camera, the denominator is fixed 0x000A. e.g.) 0x000F000A: 0x000F (means 15) / 0x0000A (means 10) = 1.5" In the case of the shutter speed is displayed as "Fraction Number" on the camera, the numerator is fixed 0x0001. 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. Real ISO value : when bits 0-23 are other than CrISO_AUTO(0xFFFFFF). 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 Only valid for models that do not support F mode. Do not use. Will be removed in the next release. This value is GetOnly. Cannot be set.
CrExposure_F_MovieOrSQMotion	F(Movie or S&Q) This value is GetOnly. Cannot be set.

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 2ics-
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 2ics-
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 2ics-
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 2ics-
CrDrive_Single_Bracket_10Ev_9pics	Single Bracket 1.0EV 9pics
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 2ics-
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 2ics-
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 2ics-
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 2ics-
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 2ics-
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 2ics-
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 2ics-
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 2ics-
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. Ex.) If min = 1000, max = 1500, step = 100, you can set 6 values of 1000, 1100, 1200, 1300, 1400, 1500 to CurrentValue. The special CurrentValue are following. - 0x0000 means less than min. - 0xFFFF means greater than max. These value is not included the value of Range. (It is only used as CurrentValue.) Note: 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. AB number is BY or AY, where Y is decimal from 0.00 to 9.00 and increments by 0.25. Ex.) B9.00(0x9C), B8.75(0x9D), ..., A8.75(0xE3), A9.00(0xE4). Note: 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. GM number is MX or GX, where X is decimal from 0.00 to 9.00 and increments by 0.25. Ex.) M9.00(0x9C), M8.75(0x9D), ..., G8.75(0xE3), G9.00(0xE4). Note: 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

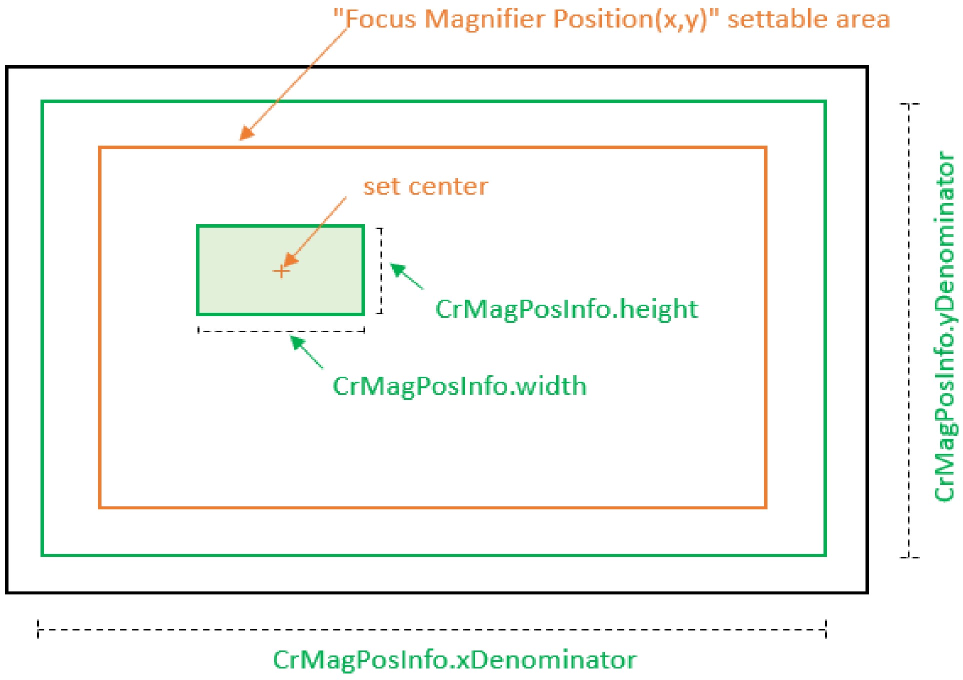
Parameter Code	Explanation
CrPriorityKey_CameraPosition	Camera position priority (Ex. Mode dial, Drive/Focus mode dial)
CrPriorityKey_PCRemote	PC Remote setting priority

CrDeviceProperty_Focus_Magnifier_Setting

Get/Set the Focus Magnifier Setting

Value	Explanation
0x0000000000000000 ～ 0xFFFFFFFFFFFFFFFF	The upper 4 bytes are the Focus Magnifier Ratio, and the lower 4 bytes are the Focus Magnifier Position(x,y). Caution : The range of focus magnifier ratio and focus magnifier position varies depending on the model and aspect ratio. [Upper 4bytes] Regarding Focus Magnifier Ratio : Select the focus magnifier ratio to be set from the focus magnifier ratio obtained by GetValues() function. Ex.) Result obtained by GetValues() function. If the camera supports OFF, x1.0, x4.0 and x8.0 as focus magnifier ratio, Result is the following. Enum value[0] = 0x00000000FFFFFFFF (means OFF) Enum value[1] = 0x0000000AFFFFFFFF (means x1.0) Enum value[2] = 0x00000028FFFFFFFF (means x4.0) Enum value[3] = 0x00000050FFFFFFFF (means x8.0) [Lower 4bytes] Regarding Focus Magnifier Position (x,y) : The upper 2 bytes are the x coordinate and the lower 2 bytes are the y coordinate. If this part is 0xFFFFFFFF, it means an invalid value. If focus magnifier position (x) is 150 and (y) is 100, set 0x00960064. 0x0096 = 0d150, 0x0064 = 0d100. The range of X is 0～639 (0x027F), and the range of Y is 0～479 (0x01DF). Frame size is acquired by CrMagPosInfo. CrMagPosInfo is in LiveViewProperty. Since this position specifies the center of the frame, the position range is more inside by half the frame size than CrMagPosInfo.xDenominator/yDenominator. Caution: 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. Note: See Tips/Trouble shooting for a detailed implementation example. Focus Magnifier Setting

Fig. Relationship between CrMagPosInfo and settable area


"Focus Magnifier Position(x,y)" settable area
set center
CrMagPosInfo.height
CrMagPosInfo.width
CrMagPosInfo.yDenominator
CrMagPosInfo.xDenominator

CrDeviceProperty_DateTime_Settings

Set the Date and Time

Parameter Code	Explanation
-	64bit value. 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. Ex.) when 1609582830 is set = 2021/01/02 10:20:30(UTC) = 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 Specify to change the focus to Near. Can be set from -1 to -7 in steps. Larger value makes the movement width larger. *1
7	max Specify to change the focus to Far. Can be set in steps of 1 to 7. Larger value makes the movement width larger. *1
1	step

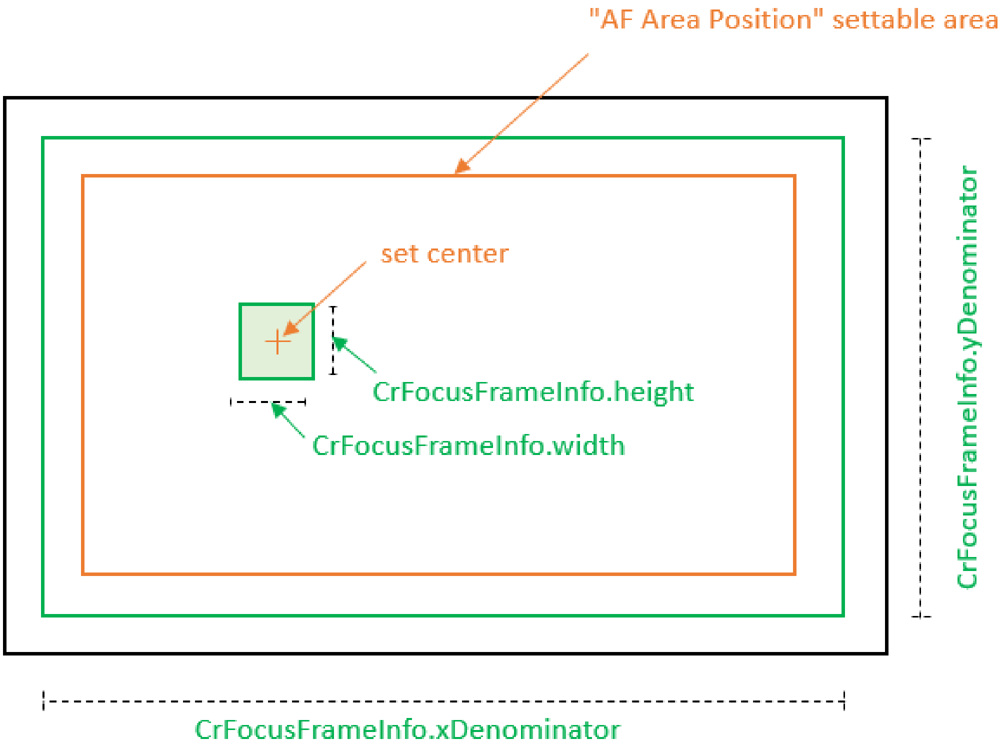
*1 : In the case of DSC-RX0M2, the movement width is fixed.

CrDeviceProperty_AF_Area_Position

Set the AF Area Position(x,y)

Value	Explanation
0x00000000～0xFFFFFFFF	Set the center position of the AF frame. The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes The range of X is 0～639 (0x027F), and the range of Y is 0～479 (0x01DF). AF frame size is acquired by CrFocusFrameInfo. CrFocusFrameInfo is in LiveViewProperty. The settable area is more inside by half the frame size than CrFocusFrameInfo.xDenominator/yDenominator. Note: 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


"AF Area Position" settable area
set center
CrFocusFrameInfo.height
CrFocusFrameInfo.width
CrFocusFrameInfo.yDenominator
CrFocusFrameInfo.xDenominator

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". Ex.) min: 1000, max: 8000, step: 200, value: 1200 (min = x1.0, max = x8.0, value = x1.2)
Variable	max	
Variable	step 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 (Negative number)	min	Zoom out (-) Default value is CrZoomOperation_Wide. 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 You can use the CrZoomOperation_Stop.
Variable (Positive number)	max	Zoom in (+) Default value is CrZoomOperation_Tele. 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

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
CrRecordingFrameRateSettingMovie_100p	100p
CrRecordingFrameRateSettingMovie_60p	60p
CrRecordingFrameRateSettingMovie_50p	50p
CrRecordingFrameRateSettingMovie_30p	30p
CrRecordingFrameRateSettingMovie_25p	25p
CrRecordingFrameRateSettingMovie_24p	24p

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

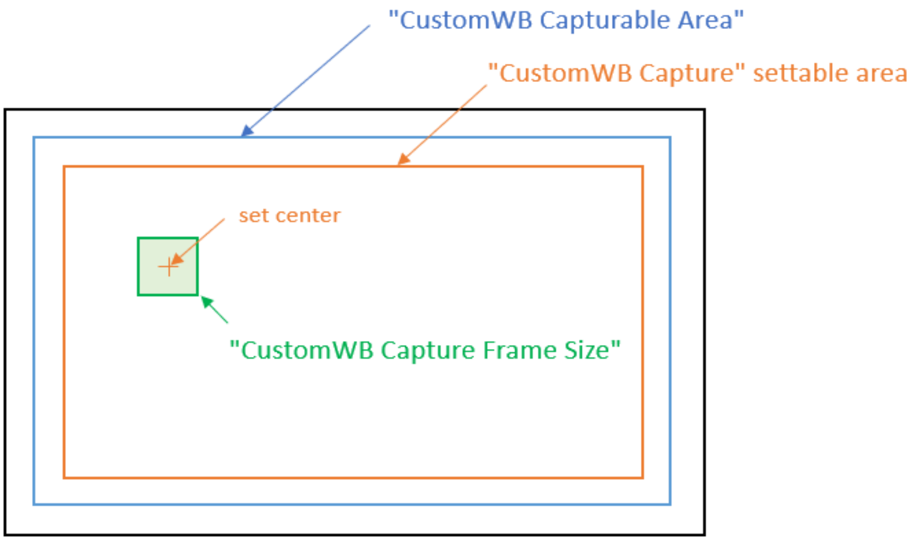
Parameter Code	Explanation
CrPropertyCustomWBCapture_Up	Up
CrPropertyCustomWBCapture_Down	Down

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 The enable range can be obtained from "Custom WB Capturable Area". The settable area is more inside by half the Frame Size than "Custom WB Capturable Area". Note: The settable range varies depending on the model and aspect setting.
0xFFFFFFFF	max	
1	step	

Fig. Relationship between capture frame size and settable position


"CustomWB Capturable Area"
"CustomWB Capture" settable area
set center
"CustomWB Capture Frame Size"

CrDeviceProperty_SnapshotInfo

Get the Shooting File Info

Value	Explanation	
0x0000	min	0x0000:transferable file doesn't exit 0x0001-0x7FFF:exist file If the value is over than 0x8001(MSB is 0b01), can 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

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

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)

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)

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

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)

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)

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 Lower 16bit is denominator, Higher 16bit is molecules. Calculate the progress rate each time. 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. The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes This value varies depends on the model and aspect setting. e.g.) 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 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

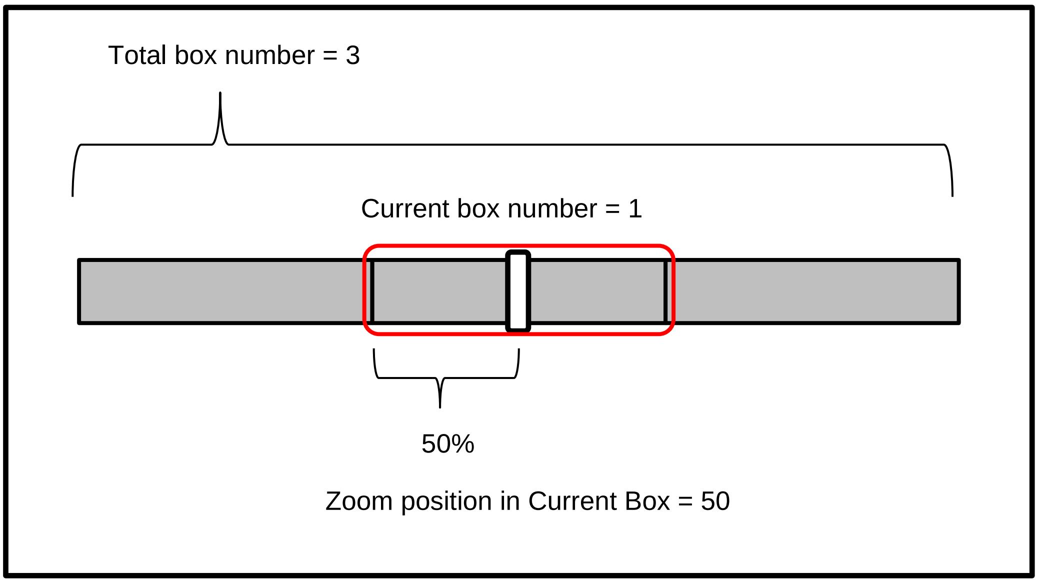
Get the Zoom Operation Enable Status

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 The current focus position, Optical Zoom position (Power Zoom lens only), and lens information are stored in the specified preset number. With CrDeviceProperty_ZoomAndFocusPosition_Load, you can get the saved preset data and restore it to the same state. Stored preset data will not be deleted even initialization. If you specify a preset number that is already in use, that preset number will be overwritten with the new preset data. e.g.) {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 Note: 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. 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. 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 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 (Negative number)	min	Zoom in speed is a positive number and zoom out speed is a negative number. Valid when CrDeviceProperty_Remocon_Zoom_Speed_Type is CrRemoconZoomSpeedType_Variable. Note: The actual zoom speed depends on the specifications of each lens model. ILME-FX3 and ILME-FX30 that not support CrDeviceProperty_Zoom_Speed_Range, but you can change the zoom speed from -8 to +8.
Variable (Positive number)	max	
1	step	

CrDeviceProperty_SdkControlMode

Get the Sdk Control Mode.

Parameter Code	Explanation
CrSdkControlMode_Remote	Remote Control Mode 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. If you do not specify openMode of the connect function, connect in this mode.
CrSdkControlMode_ContentsTransfer	Contents Transfer Mode 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 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 Content handle during transfer processing
31-0bit	0-100 Transfer progress rate. Unit is percent(%) Content with a large file size is acquired in multiple steps. The acquisition time changes depending on the size of the file size. 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 Set the EI value, The set value varies depending on the model and the setting status of the camera. See GetDisplayStringList() for display character string and highlight latitude list associated with EI. Ex.) If setting with "200EI / 4.0E", set 0x00C8.

CrDeviceProperty_BaseLookValue

Get/Set the BaseLook Value

Parameter Code		Explanation	
15-8bit		Kind	16bit value that combines Kind(upper 8bit) and Index (lower 8bit) Ex.) 0x0003 = 3(Preset) 0x0108 = 8(User) It may increase or decrease because it varies depending on the model and setting status. See the GetDisplayStringList() for display character string.
	CrBaseLookValue_Preset(0x00)		
	CrBaseLookValue_User(0x01)		
0-7bit		Index	

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. 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.
	
	values[0] = 0x00000002 (Display All Information)
	values[1] = 0x00000004 (Histogram)
	values[2] = 0x00000008 (Level)
	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. 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. Note: Not all items can be disabled. Be sure to set one or more items(bit position) to enable(turn on the bit). 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 Do not save the CurrentValue of CrDeviceProperty_ImageID_Num to the Exif tag of the image.
CrImageIDNumSetting_On	ON Save the CurrentValue of CrDeviceProperty_ImageID_Num to the Exif tag of the image. Caution: 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. Save the value in the Exif tag of the image file only if CrDeviceProperty_ImageID_Num_Setting is CrImageIDNumSetting_On. 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. Note: 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. If blank ("") is set, Exif tags are not save in the image. Note: 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. Real ISO value : when bits 0-23 are other than CrISO_AUTO(0xFFFFFF). 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. If current value is CrFocalDistance_Infinity(0xFFFFFFFF), ∞. e.g.) 0x00005014 = 20500 /1000 = 20.5 meter 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. If current value is CrFocalDistance_Infinity(0xFFFFFFFF), ∞. e.g.) 0x00005014 = 20500 /1000 = 20.5 feet 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. The resolution of the CurrentValue is the step value. The CurrentValue increases or decreases with each step value. Ex.) 0x000004B0 = 1200 /1000 = x1.2 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. min/max/CurrentValue should be set in units of “step”. Ex.) min: 18000, max: 55000, step: 1000, value: 20000 (min = 18mm, max = 55mm, value = 20mm) The maximum value as a protocol is 4294967 mm. Note: 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 （0x00000000 ～0xFFFFFFFF）	Recording resolution(Width , Height) The "Width" is set in the upper two bytes and the "Height" is set in the lower two bytes if resolution (Width) is 1920, (Height) is 1080, set 0x07800438. 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 （0x00000000 ～0xFFFFFFFF）	Recording resolution(Width , Height) The "Width" is set in the upper two bytes and the "Height" is set in the lower two bytes if resolution (Width) is 1920, (Height) is 1080, set 0x07800438. 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
CrRecordingFrameRateProxySettingMovie_50p	50p
CrRecordingFrameRateProxySettingMovie_25p	25p
CrRecordingFrameRateProxySettingMovie_24p	24p
CrRecordingFrameRateProxySettingMovie_23_98p	23.98p
CrRecordingFrameRateProxySettingMovie_29_97p	29.97p
CrRecordingFrameRateProxySettingMovie_59_94p	59.94p

CrDeviceProperty_MovieShootingMode

Get/Set the Movie Shooting Mode

For ILME-FX6, only gets are supported.

Parameter Code	Explanation
CrMovieShootingMode_CineEI	Cine EI
CrMovieShootingMode_Custom	Custom

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. This device property is used to update CrDeviceProperty_Colortemp. 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. 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. Note: 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. 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. Can be set within the range of CrDeviceProperty_Focus_Speed_Range. Ex.) SetValue = 1 ： Tele focus (focus speed=1) SetValue = -3 ： Wide focus (focus speed=3) SetValue = 0 ： Stop focus

CrDeviceProperty_Focus_Speed_Range

Get the Focus Speed Range

Parameter Code	Explanation	
Variable (Negative number)	min	A value that can be used for Focus Operation. For example, when min is -5 and max is +5, it means that the focus drive speed can be specified in 5 steps. The higher the number, the faster the focus drive speed. The CurrentValue of this device property is always zero.
Variable (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. If CrDeviceProperty_ShutterECSNumberStep is operated while the CurrentValue is min or max, the CurrentValue will not be changed. Note: 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. 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 (0x00～0xFF)	Assign a certain function to CrDeviceProperty_AssignableButton1 so that the function can be executed by button operation. 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. See "Get the menu display string"

CrDeviceProperty_ButtonAssignmentAssignable2

Get/Set the Button Assignment Assignable 2

Parameter Code	Explanation
Variable (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 (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 (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 (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 (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 (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 (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 (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 (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. 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 e.g.) 0x0002BF20 = 180000 /1000 = 180 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. 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. 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. 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. 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. 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. 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. 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. 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: The range value may changes depending on the model.
Variable	max	
Variable	step	

CrDeviceProperty_AFSubjShiftSens

Get/Set AF Subj Shift Sens

Parameter Code	Explanation	
Variable	min	Note: The range value may changes 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. This property can only be executed if CrDeviceProperty_RemoteTouchOperationEnableStatus is Enable. The x coordinate is set in the upper two bytes and the y coordinate is set in the lower two bytes The range of X is 0～639 (0x027F), and the range of Y is 0～479 (0x01DF).
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) 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. 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. 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. 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. Note: Can be converted from normalized values to Focus Position (meters/feet) using data taken with GetLensInformation(). 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. 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

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, 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.

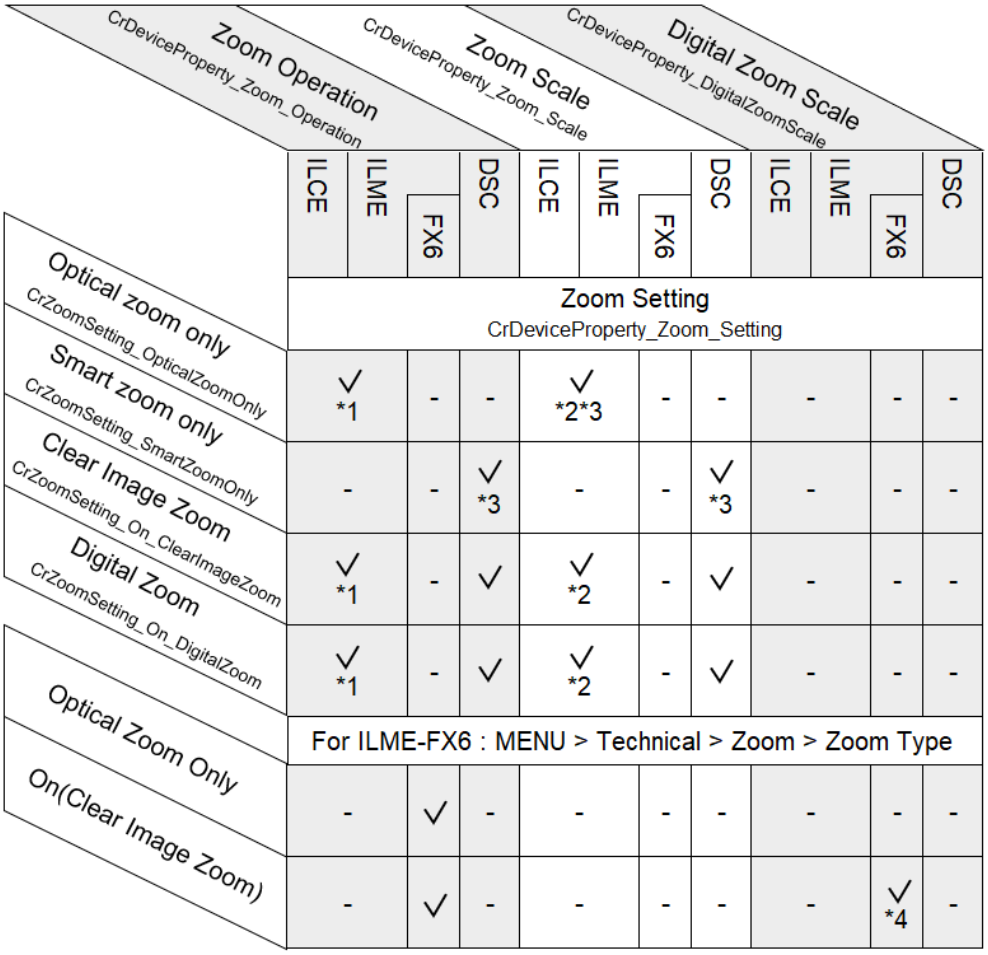
If you want to initialize it, change Still Image Save Destination to Camera or Host Device and Camera.

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.



	Zoom Operation CrDeviceProperty_Zoom_Operation				Zoom Scale CrDeviceProperty_Zoom_Scale				Digital Zoom Scale CrDeviceProperty_DigitalZoomScale			
	ILCE	ILME	FX6	DSC	ILCE	ILME	FX6	DSC	ILCE	ILME	FX6	DSC
	Zoom Setting CrDeviceProperty_Zoom_Setting											
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		-	√	-		-	-
	For ILME-FX6 : MENU > Technical > Zoom > Zoom Type											
Optical Zoom Only	-		√	-	-		-	-	-		-	-
On(Clear Image Zoom)	-		√	-	-		-	-	-		√ *4	-

*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. 0x00000020 and 0x00000040 are exclusive. 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);

}
```

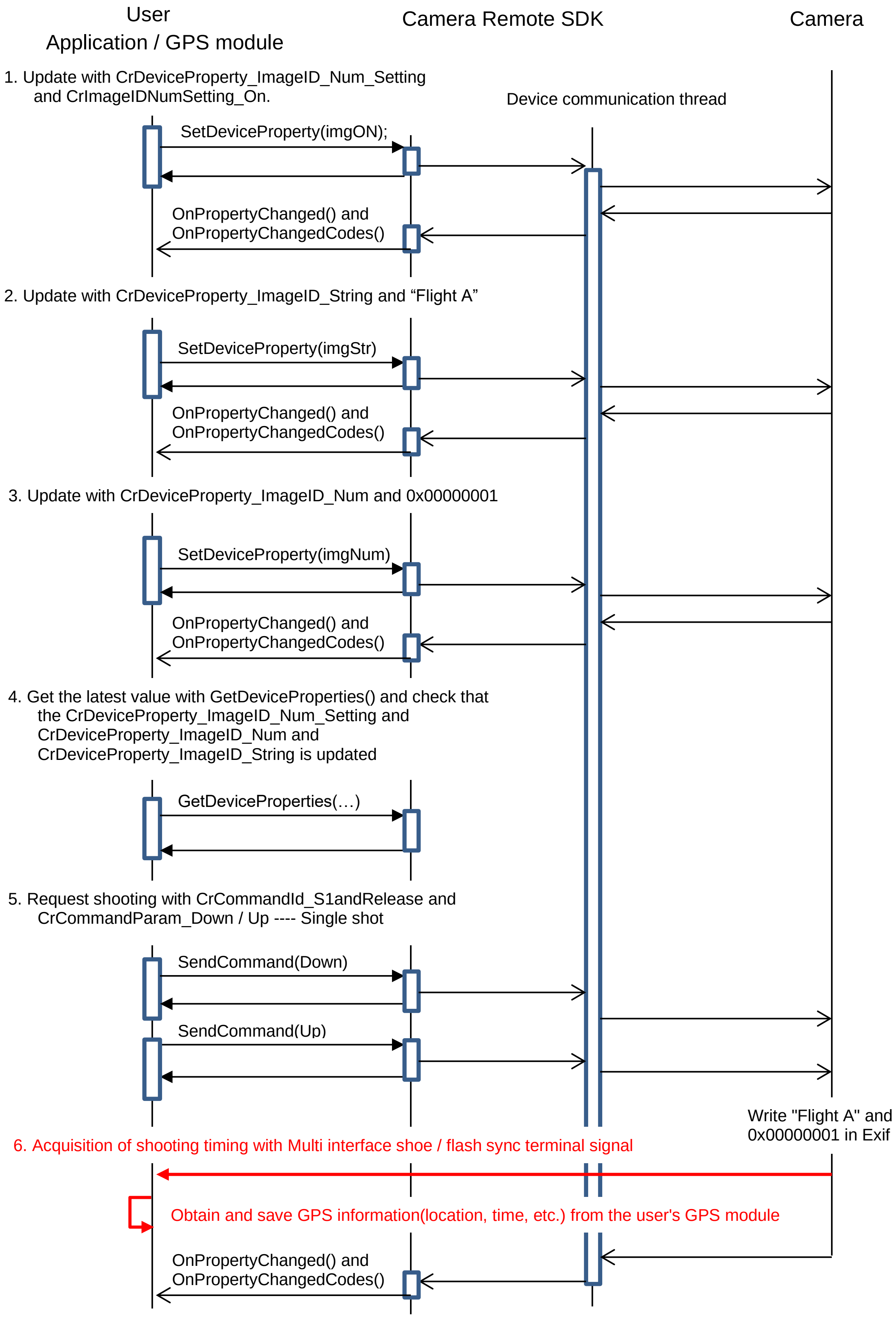
The information retrieved by GetLensInformation() can be used to know the Focus position (meter/feet).

Example:

```
                // Example of a case where CrDevicePropertyFollowFocusPositionCurrentValue 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



1. Update with CrDeviceProperty_ImageID_Num_Setting and CrImageIDNumSetting_On.

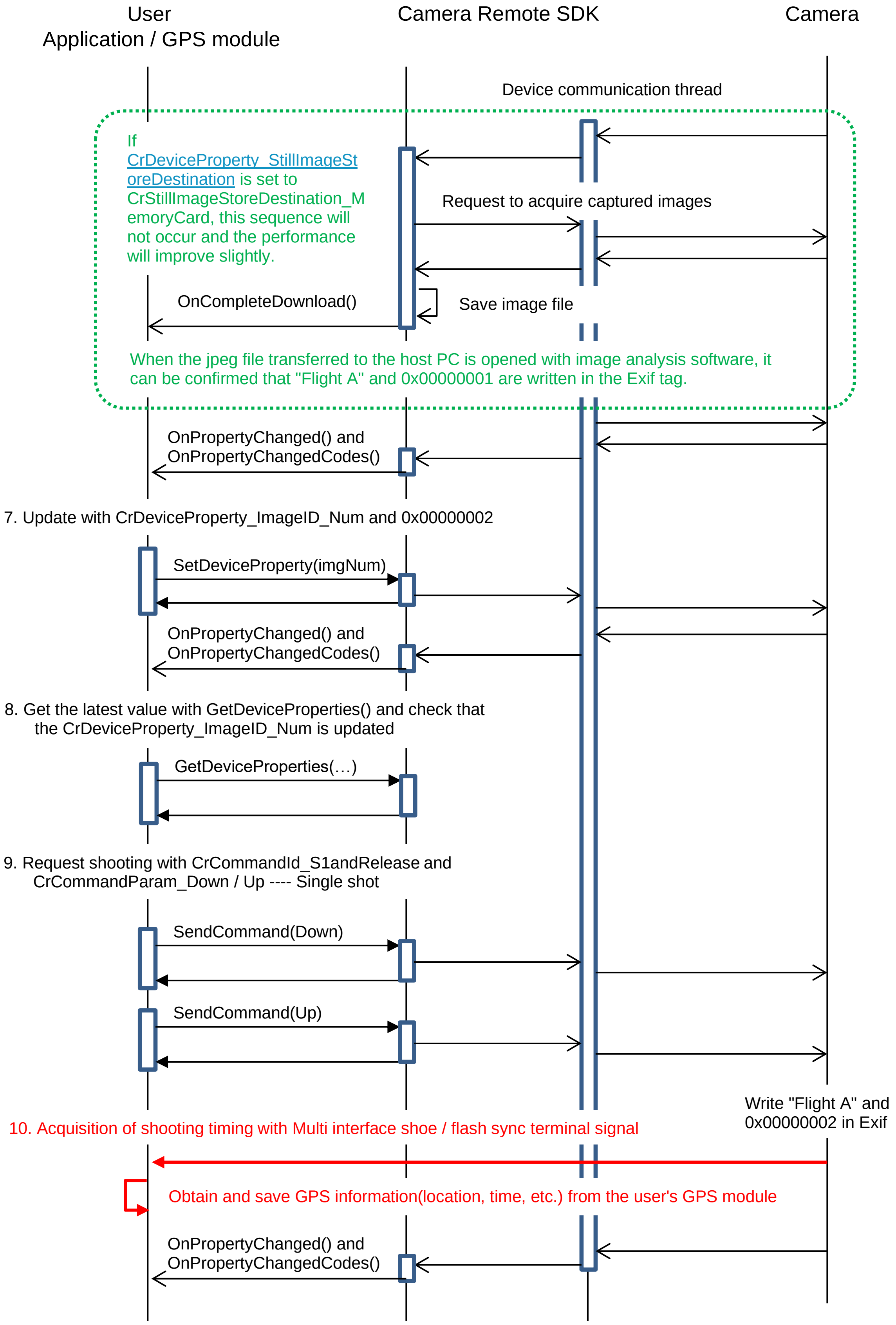
2. Update with CrDeviceProperty_ImageID_String and “Flight A”

3. Update with CrDeviceProperty_ImageID_Num and 0x00000001

4. Get the latest value with GetDeviceProperties() and check that the CrDeviceProperty_ImageID_Num_Setting and CrDeviceProperty_ImageID_Num and CrDeviceProperty_ImageID_String is updated

5. Request shooting with CrCommandId_S1andRelease and CrCommandParam_Down / Up ---- Single shot

6. Acquisition of shooting timing with Multi interface shoe / flash sync terminal signal



If CrDeviceProperty_StillImageStoreDestination is set to CrStillImageStoreDestination_MemoryCard, this sequence will not occur and the performance will improve slightly.

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.

7. Update with CrDeviceProperty_ImageID_Num and 0x00000002

8. Get the latest value with GetDeviceProperties() and check that the CrDeviceProperty_ImageID_Num is updated

9. Request shooting with CrCommandId_S1andRelease and CrCommandParam_Down / Up ---- Single shot

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

More information

Trademarks and acknowledgements