

Canon EOS Digital SDK

# EDSDK for RAW Develop 13.19.0 Programming Reference

02/05/2025

## History

Version	Date	Revised page(s)	Reason and content of revision	Reviser
13.10.21	06/21/2019		First release	
13.10.30	09/10/2019		* Added support for the EOS M6 Mark II / EOS 90D	
13.12.1	01/09/2020		* Added support for the EOS-1D X Mark III / EOS M200 * Added support for Mac OSX 10.15	
13.12.31	06/30/2020		* Added support for the EOS R5 / EOS R6 / EOS Ra / EOS Kiss X10i / EOS Rebel T8i / EOS 850D * Stopping support OS * Mac OS 10.12	
13.13.0	09/30/2020		* Added support for the EOS Kiss M2 / EOS M50 Mark II	
13.14.0	12/02/2021		* Added support for the EOS R3	
13.15.20	07/21/2022		* Added support for the EOS R7 / EOS R10	
13.16.0	11/09/2022		* Added support for the EOS R6 Mark II	
13.16.10	12/23/2022		* Added support for the EOS R8 / EOS R50	
13.17.0	05/25/2023		* Added support for the EOS R100	
13.18.0	02/05/2024		* "Supported Enviroments " is updated.	
13.18.30	07/26/2024		* Added support for the EOS R5 Mark II	
13.18.40	09/26/2024		* Added support for the EOS R1	
13.19.0	02/05/2025		* Added support for the EOS R50V	

Revision History/Date	Corrections	Reviser	Remarks

## Table of Contents

<b>1. INTRODUCTION.....</b>	<b>5</b>
1.1 Basic Topics.....	5
1.2 Supported Environments.....	5
1.2.1 EDS SDK for Windows.....	5
1.2.2 EDS SDK for macOS.....	5
1.3.1 Supported Cameras and OS Compatibility Table.....	6
1.4 Installing EDS SDK for RAW Develop.....	9
1.4.1 Including Header Files.....	9
1.4.2 Linking the Library.....	9
1.4.3 Executing the EDS SDK for RAW Develop Client Application.....	10
<b>2. OVERVIEW.....</b>	<b>11</b>
2.1 Library Modules.....	11
2.2 Handling Image Objects.....	12
2.2.1 Overview.....	12
2.2.2 Getting and Setting Properties.....	12
<b>3. API REFERENCE.....</b>	<b>13</b>
3.1 API Details.....	13
3.1.1 EdsGetImageInfo.....	14
3.1.2 EdsGetImage.....	15
3.1.3 EdsSaveImage.....	17
3.1.4 EdsCacheImage.....	18
3.2 EDS Error Lists.....	19
<b>4. ASYNCHRONOUS EVENTS.....</b>	<b>20</b>
<b>5. PROPERTIES.....</b>	<b>21</b>
5.1 Property Lists.....	21
5.2 Property Details.....	22
5.2.1 kEdsPropID_AtCapture_Flag.....	23
5.2.2 kEdsPropID_Orientation.....	24
5.2.3 kEdsPropID_DigitalExposure.....	25
5.2.4 kEdsPropID_WhiteBalance.....	26
5.2.5 kEdsPropID_ColorTemperature.....	27
5.2.6 kEdsPropID_WhiteBalanceShift.....	27
5.2.7 kEdsPropID_ClickWBPoint.....	28
5.2.8 kEdsPropID_WBCoeffs.....	28
5.2.9 kEdsPropID_Linear.....	29
5.2.10 kEdsPropID_ColorSpace.....	29
5.2.11 kEdsPropID_PictureStyle.....	30
5.2.12 kEdsPropID_PictureStyleDesc.....	31

Revision History/Date	Corrections	Reviser	Remarks

6. APPENDIX..... 32

6.1 Data Types Used by the APIs .....32

6.1.1 EdsSaveImageSetting.....32

Revision History/Date		Corrections	Reviser	Remarks

## 1. Introduction

This document explains EDS SDK APIs for RAW Develop. So this document mainly explains functions that are different from the EDS SDK API document and additional functions. For camera control functions, please refer to EDS SDK API document.

EDS SDK APIs for RAW Develop provides an interface for accessing image data shot using a Canon digital camera. Using EDS SDK APIs for RAW Develop allows users to implement the following functions in software.

- Allows RAW images to be processed and saved in JPEG format.

### 1.1 Basic Topics

EDS SDK provides a C language interface for accessing Canon digital cameras and data created these cameras. EDS SDK is designed to provide standard methods of accessing different camera models and their data. Using EDS SDK allows users to implement Canon digital camera features in software.

There are two versions of EDS SDK. One runs under a Windows environment, while the other runs under a Macintosh environment.

### 1.2 Supported Environments

EDS SDK can be used on system configurations such as shown in the table below.

#### 1.2.1 EDS SDK for Windows

Checked with the environment on Windows 10,11 (64bit/32bit)

#### 1.2.2 EDS SDK for macOS

Checked with the environment on macOS v13-15 (64bit)

#### Notes:

It does not support native execution on a mac with Apple silicon.

In macOS 14.0-14.1, connection failures occur. Please use macOS 14.2 or later.

Revision History/Date	Corrections	Reviser	Remarks

## 1.3 Supported Cameras

### 1.3.1 Supported Cameras and OS Compatibility Table

The following models are supported as of February 2025.

Supported Cameras	Supported OS			Limitations
	Windows		Macintosh	
	64bit	32bit	64bit	
EOS R50V	✓	-	✓	*1*2
EOS R1	✓	-	✓	*1*2
EOS R5 Mark II	✓	-	✓	*1*2
EOS R100	✓	-	✓	*1*2
EOS R50	✓	-	✓	*1*2
EOS R7	✓	-	✓	*1*2
EOS R6 Mark II	✓	-	✓	*1*2
EOS R10	✓	-	✓	*1*2
EOS R7	✓	-	✓	*1*2
EOS R3	✓	-	✓	*1*2
EOS Kiss M2 / EOS M50 Mark II	✓	-	✓	*1
EOS R5	✓	-	✓	*1*2
EOS R6	✓	-	✓	*1*2
EOS Kiss X10i / EOS Rebel T8i / EOS 850D	✓	-	✓	*1
EOS Ra	✓	-	✓	*1
EOS-1D X Mark III	✓	-	✓	*1*2
EOS M200	✓	-	✓	*1
EOS M6 Mark II	✓	-	✓	*1
EOS 90D	✓	-	✓	*1
PowerShot G7X Mark III	✓	✓	✓	*1
PowerShot G5X Mark II	✓	✓	✓	*1
EOS Kiss X10 / EOS Rebel SL3 / EOS 250D / EOS 200D II	✓	✓	✓	*1
EOS RP	✓	✓	✓	*1
PowerShot SX70 HS	✓	✓	✓	*1
EOS R	✓	✓	✓	*1
EOS Kiss M / EOS M50	✓	✓	✓	*1
EOS Kiss X90 / EOS REBEL T7 / EOS 2000D / EOS 1500D	✓	✓	✓	*1
EOS REBEL T100/EOS 4000D / EOS 3000D	✓	✓	✓	*1
EOS M100	✓	✓	✓	*1
EOS 6D Mark II	✓	✓	✓	*1
EOS Kiss X9 / EOS Rebel SL2 /	✓	✓	✓	*1

Revision History/Date		Corrections	Reviser	Remarks

EOS 200D				
EOS Kiss X9i / EOS Rebel T7i / EOS 800D	✓	✓	✓	*1
EOS 9000D / EOS 77D	✓	✓	✓	*1
EOS M6	✓	✓	✓	*1
EOS M5	✓	✓	✓	*1
EOS 5D Mark IV	✓	✓	✓	*1
EOS-1D X Mark II	✓	✓	✓	*1
EOS 80D	✓	✓	✓	*1
EOS Kiss X80 / EOS Rebel T6 / EOS 1300D	✓	✓	✓	*1
EOS M10	✓	✓	✓	*1
EOS 5DS	✓	✓	✓	*1
EOS 5DS R	✓	✓	✓	*1
EOS 8000D / EOS REBEL T6sEOS 760D	✓	✓	✓	*1
EOS Kiss X8i / EOS REBEL T6i / EOS 750D	✓	✓	✓	*1
EOS M3	✓	✓	✓	*1
EOS 7D Mark II	-	✓	-	-
EOS Kiss X70/EOS 1200D/EOS REBEL T5/EOS Hi	-	✓	-	-
EOS M2	-	✓	-	-
EOS 70D	-	✓	-	-
EOS Kiss X7 / EOS 100D / EOS REBEL SL1	-	✓	-	-
EOS Kiss X7i / EOS 700D / EOS REBEL T5i	-	✓	-	-
EOS-1D C	-	✓	-	-
EOS 6D	-	✓	-	-
EOS M	-	✓	-	-
EOS Kiss X6i / EOS 650D / EOS REBEL T4i	-	✓	-	-
EOS-1D X	-	✓	-	-
EOS 5D Mark III	-	✓	-	-
EOS Kiss X50 / EOS REBEL T3 / EOS 1100D	-	✓	-	-
EOS Kiss X5 / EOS REBEL T3i / EOS 600D	-	✓	-	-
EOS 60D	-	✓	-	-
EOS Kiss X4 / EOS REBEL T2i / EOS 550D	-	✓	-	-
EOS-1D Mark IV	-	✓	-	-
EOS 7D	-	✓	-	-
EOS Kiss X3 / EOS REBEL T1i / EOS 500D	-	✓	-	-

Revision History/Date	Corrections	Reviser	Remarks

EOS 5D Mark II	-	✓	-	-
EOS 50D	-	✓	-	-
EOS DIGITAL REBEL XS / 1000D/ KISS F	-	✓	-	-
EOS DIGITAL REBEL Xsi / 450D / Kiss X2	-	✓	-	-
EOS-1Ds Mark III	-	✓	-	-
EOS 40D	-	✓	-	-
EOS-1D Mark III	-	✓	-	-

**Notes:** \*1 The following properties are not supported.

- kEdsPropID\_WBCoeffs

\*2 HEIF format is not supported.

Revision History/Date		Corrections	Reviser	Remarks



## 1.4 Installing EDSDK for RAW Develop

### 1.4.1 Including Header Files

The following files are required in order to use the EDSDK for RAW Develop using C/C++ language.

EDSDK.h, EDSDKTypes.h, EDSDKErrors.h

#### Windows:

Be sure to copy the three header files listed above into the header access folder of the development environment.

Be sure to add them to the application project workspace.

\*Since these are C language header files, it is necessary to provide header files depending on the programming language.

#### Macintosh:

Be sure to include the three header files listed above.

### 1.4.2 Linking the Library

After header files are included, it is necessary to link the EDSDK library as described below.

#### Windows:

There are two methods of linking EDSDK: one where EDSDK.lib files are copied to the folder specified by a development environment library path and EDSDK.lib is specified as an import module, and another where EDSDK.dll is loaded by the LoadLibrary function.

When loading EDSDK.dll, get pointers to each EDSDK function using the GetProcAddress function and assign them to function pointer variables. When calling each EDSDK function, make the call via the function pointer variable obtained here.

#### Macintosh:

Add EDSDK.framework and DPP.framework to Groups & Files.

Revision History/Date		Corrections	Reviser	Remarks

### 1.4.3 Executing the EDSDK for RAW Develop Client Application

**Windows:**

All EDSDK for RAW Develop library files are required in order to execute an EDSDK for RAW Develop client application.

All of the modules in the DLL folder must be copied into the same folder where the EDSDK for RAW Develop client application is in.

**Notes:**

Do not copy the collection of EDSDK for RAW Develop library files to the system folder or extension folder.

**Macintosh:**

Place EDSDK.framework in an application directory such as Contents/frameworks/.

It is also possible to load “EDSDK.framework” as a source file. The following code has been written as an Objective-C source.

```
-(id)init {
    // START to Load EDSDK.framework -----
    NSString *symName = @"EDSDK.framework" ;
    int i;
    NSArray *array = [NSBundle allFrameworks];
    void *symData = NULL;

    for (i = 0; symData == NULL && i < [array count]; i++) {
        NSBundle *framework = [array objectAtIndex:i];
        NSString *bundleID = [framework bundleIdentifier];
        if (bundleID) {
            CFBundleRef bundle = CFBundleGetBundleWithIdentifier((CFStringRef) bundleID);
            if (bundle) {
                symData = CFBundleGetFunctionPointerForName(bundle, (CFStringRef) symName);
            }
        }
    }
    // END of Loading EDSDK.framework -----

    EdsError err = EDS_ERR_OK ;
    err = EdsInitializeSDK() ;
}
```

**Notes:**

Do not copy the EDSDK framework file to the system folder.

Revision History/Date	Corrections	Reviser	Remarks

## 2. Overview

This section introduces overview of EDS SDK APIs for RAW Develop.

For overview of EDS SDK APIs that are not only for RAW Develop, please refer to EDS SDK API document.

### 2.1 Library Modules

The following figure shows the module configuration of EDS SDK for RAW Develop.

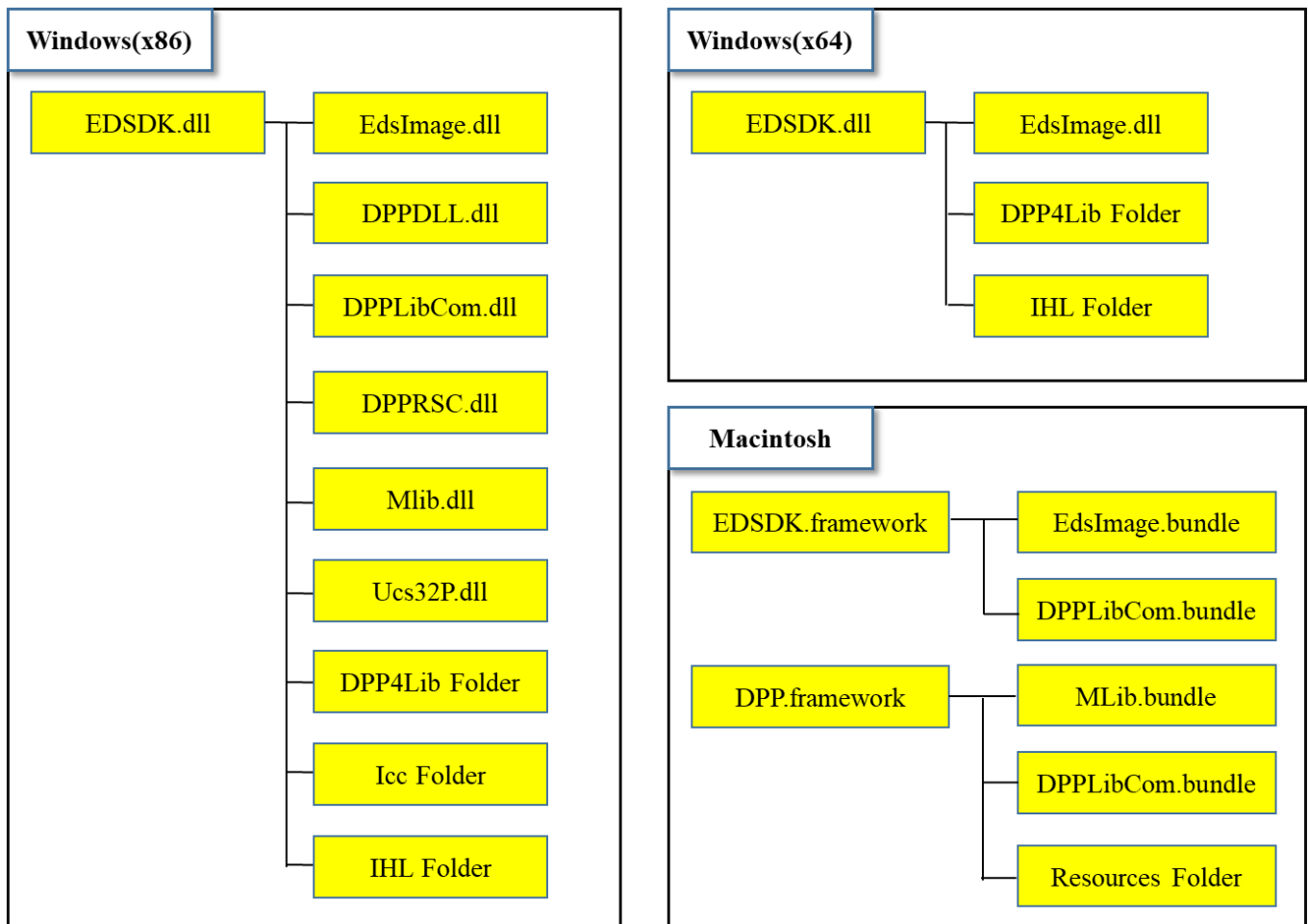


Figure 2-1 Library Module Configuration

Revision History/Date		Corrections	Reviser	Remarks

## 2.2 Handling Image Objects

### 2.2.1 Overview

It is possible to process an image under conditions other than those at the time the image was shot by setting processing-related properties such as the white balance and picture style using **EdsSetPropertyData** if the image object is RAW.

### 2.2.2 Getting and Setting Properties

When processing is carried out using **EdsGetImage** or **EdsSaveImage** by setting properties for the image object, the specified property settings will be reflected in the generated JPEG. Note, however, that changes to properties will not be reflected in the source image stored by **EdsImageRef**.

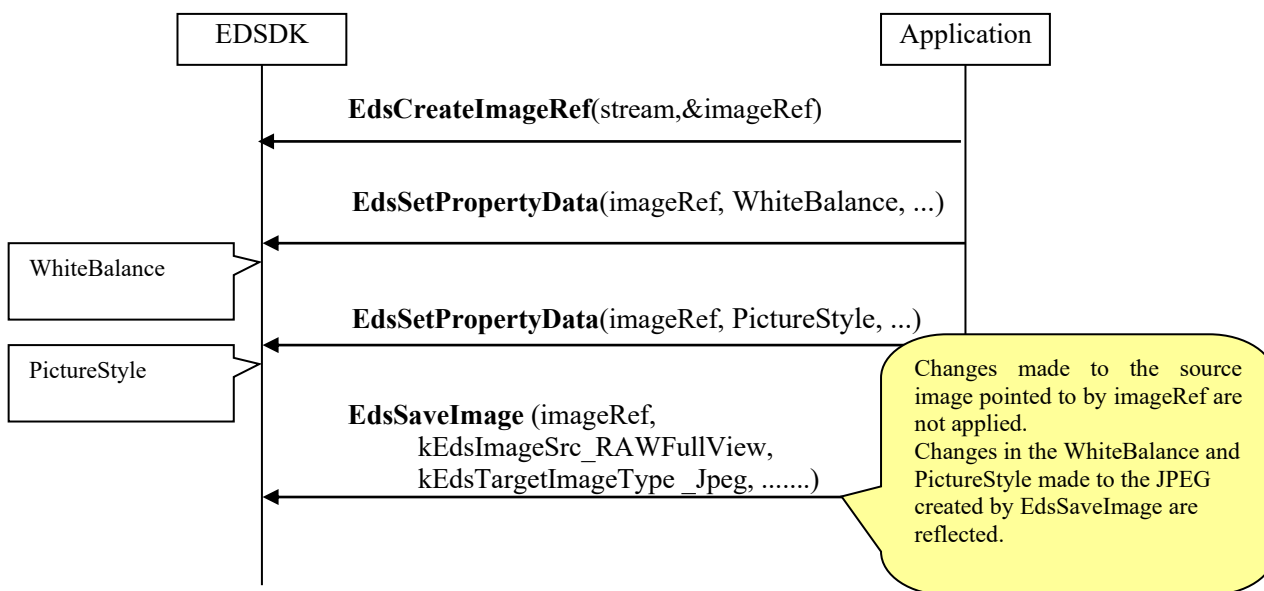


Figure 2-21 Setting Properties Reflected in the Resulting Processed Image

Revision History/Date	Corrections	Reviser	Remarks

### 3. API Reference

This section introduces EDS SDK APIs for RAW Develop.

For EDS SDK APIs that are not only for RAW Develop, please refer to EDS SDK API document.

#### 3.1 API Details

API specifications are explained in the following format.

##### Description

Indicates the main API function.

##### Syntax

EdsError EdsXXXXX( EdsUInt32 **in**XXXX, EdsBaseRef \***out**XXX );

Indicates the syntax for calling the API.

##### Parameters

Explains each argument in the syntax individually.

In the syntax, argument names in the format **in**XXX represent arguments for which you enter values. Argument names in the format **out**XXX represent arguments with values set by the libraries (that is, passed by reference). Before calling APIs, you must prepare variables for storing the data to be retrieved.

##### Return Values

Explains API return values.

##### See Also

Indicates information related to the API.

##### Note

Considerations when using the API.

##### Example

Sample code.

Revision History/Date		Corrections	Reviser	Remarks

### 3.1.1 EdsGetImageInfo

#### Description

Gets image information from a designated image object.

Here, image information means the image width and height, number of color components, resolution, and effective image area.

#### Syntax

```
EdsError EdsGetImageInfo(
    EdsImageRef inImageRef, EdsImageSource inImageSource,
    EdsImageInfo* outImageInfo )
```

#### Parameters

**inStreamRef**

Designate the object for which to get image information.

**inImageSource**

Of the various image data items in the image file, designate the type of image data representing the information you want to get. Designate the image as defined in Enum EdsImageSource.

Enum EdsImageSource      <defined location>EDSDKTypes.h

Value	Description
kEdsImageSrc_FullView	The image itself (a full-sized image)
kEdsImageSrc_Thumbnail	A thumbnail image
kEdsImageSrc_Preview	A preview image
kEdsImageSrc_RAWThumbnail	A RAW thumbnail image
kEdsImageSrc_RAWFullView	A RAW full-sized image

**outImageInfo**

Stores the image data information designated in inImageSource.

EdsImageInfo constituent elements	Type	Description
width	EdsUInt32	Width (in pixels)
height	EdsUInt32	Height (in pixels)
numOfComponents	EdsUInt32	Number of color components
componentDepth	EdsUInt32	Resolution (8-bit or 16-bit) Note: Image files may contain image data of mixed resolutions.
effectiveRect	EdsRect	Effective image area (This means the area excluding the black bands on the top and bottom of the thumbnail image.)
Reserved	EdsUInt32	Reserved

#### Return Values

Returns EDS\_ERR\_OK if successful. In other cases, see the EDS Error Lists.

#### See Also

- Related APIs  
EdsCreateImageRef and EdsGetImage
- For information on data types of the EDSDK, see "Data Types Used by the APIs" in the Appendix.

Revision History/Date	Corrections	Reviser	Remarks

### 3.1.2 EdsGetImage

#### Description

Gets designated image data from an image file, in the form of a designated rectangle.  
Returns uncompressed results for JPEG compressed images and processed results in the designated pixel order (RGB, Top-down BGR, and so on) for RAW images. Additionally, by designating the input/output rectangle, it is possible to get reduced, enlarged, or partial images. However, because images corresponding to the designated output rectangle are always returned by the SDK, the SDK does not take the aspect ratio into account. To maintain the aspect ratio, you must keep the aspect ratio in mind when designating the rectangle.

#### Syntax

```
EdsError EdsGetImage(
    EdsImageRef      inImageRef,
    EdsImageSource    inImageSource,
    EdsTargetImageType inImageType,
    EdsRect           inSrcRect,
    EdsSize           inDstSize,
    EdsStreamRef      outStreamRef
);
```

#### Parameters

**inImageRef**

Designate the image object for which to get the image data.

**inImageSource**

Designate the type of image data to get from the image file (thumbnail, preview, and so on).  
Designate values as defined in Enum EdsImageSource.

Enum EdsImageSource <defined location>EDSDKTypes.h

Value	Description
kEdsImageSrc_FullView	The image itself (a full-sized image)
kEdsImageSrc_Thumbnail	A thumbnail image
kEdsImageSrc_Preview	A preview image (displayed on the back screen of the camera)
kEdsImageSrc_RAWThumbnail	A RAW thumbnail image
kEdsImageSrc_RAWFullView	A RAW full-sized image

**inImageType**

Designate the output image type. Because the output format of EdsGetImage may only be RGB, only **kEdsTargetImageType\_RGB** or **kEdsTargetImageType\_RGB16** can be designated.  
However, image types exceeding the resolution of inImageSource cannot be designated.

Example: Suppose the source image resolution (componentDepth) retrieved by means of **EdsGetImageInfo()** is 8 bits

- The resolution that can be retrieved by means of EdsGetImage () is also 8 bits
- Thus, only **kEdsTargetImageType\_RGB** is available.

EdsTargetImageType <defined location>EDSDKTypes.h

Value	Description
kEdsTargetImageType_RGB	8-bit RGB, chunky format
kEdsTargetImageType_RGB16	16-bit RGB, chunky format

Revision History/Date	Corrections	Reviser	Remarks

inSrcRect

Designate the coordinates and size of the rectangle to be retrieved (processed) from the source image.

inDstSize

Designate the rectangle size for output.

outStreamRef

Designate the memory or file stream for output of the image.

### Return Values

Returns EDS\_ERR\_OK if successful. In other cases, see the EDS Error Lists.

### See Also

- Related APIs  
EdsCreateImageRef and EdsGetImageInfo

### Note

- To maintain the aspect ratio, you must keep the aspect ratio in mind when designating a rectangle.
- In calculating the data size of the output file, the original image data resolution is not used. Instead, the resolution of the image type designated by inImageType is used. For example, the calculation for kEdsTargetImageType\_RGB is 3 (R, G, and B) x 8 (resolution) x width x height ÷ 8 (bytes). Similarly, kEdsTargetImageType\_RGB16 is calculated by 3 x 16 x width x height ÷ 8 (bytes).

Revision History/Date		Corrections	Reviser	Remarks



### 3.1.3 EdsSaveImage

#### Description

Saves as a designated image type after RAW processing.

#### Syntax

```
EdsError  EdsSaveImage(
                EdsImageRef          inImageRef,
                EdsTargetImageType    inImageType,
                EdsSaveImageSetting    inSaveSetting,
                EdsStreamRef           outStreamRef
            );
```

#### Parameters

**inImageRef**

Designate the image object for which to produce the file.

**inImageType**

Designate the image type to produce. Designate the following image types.

Enum EdsTargetImageType <defined location>EDSDKTypes.h

Value	Description
kEdsTargetImageType_Jpeg	JPEG
kEdsTargetImageType_TIFF	8-bit TIFF
kEdsTargetImageType_TIFF16	16-bit TIFF

**inSaveSetting**

Designate saving options, such as JPEG quality.

EdsSaveImageSetting <defined location>EDSDKTypes.h

EdsSaveImageSetting constituent elements	Type	Description
JPEGQuality	EdsUInt32	Image quality for JPEG compression 1 (rough) to 10 (fine)
iccProfileStream	EdsStreamRef	ICC profile stream
reserved	EdsUInt32	Reserved

**outStreamRef**

Specifies the output file stream. The memory stream cannot be specified here.

#### Return Values

Returns EDS\_ERR\_OK if successful. In other cases, see the EDS Error Lists.

#### See Also

- Related APIs  
EdsCreateFileStream, EdsCreateFileStreamEx, EdsRead, and EdsWrite
- For information on data types of the EDS SDK, see "Data Types Used by the APIs" in the Appendix.

#### Note

Revision History/Date	Corrections	Reviser	Remarks

### 3.1.4 EdsCacheImage

#### Description

Switches a setting on and off for creation of an image cache in the SDK for a designated image object during extraction (processing) of the image data. Creating the cache increases the processing speed, starting from the second time.

#### Syntax

```
EdsError  EdsCacheImage(
                                EdsImageRef  inImageRef,
                                EdsBool       inUseCache
                            );
```

#### Parameters

inImageRef

Designate the image object.

inUseCache

TRUE: Image cache ON

FALSE: Image cache OFF

#### Return Values

Returns EDS\_ERR\_OK if successful. In other cases, see the EDS Error Lists.

#### See Also

- Related APIs  
EdsGetImage and EdsSaveImage

#### Note

- If the image cache is on, a corresponding amount of resources are consumed. If fast processing is not required, use the EDSDK with the cache off.

Revision History/Date		Corrections	Reviser	Remarks

### 3.2 EDS Error Lists

There is no change from EDSDK API document in this chapter.  
Please refer to EDSDK API document.

Revision History/Date		Corrections	Reviser	Remarks

## 4. Asynchronous Events

There is no change from EDSK API document in this chapter.  
Please refer to EDSK API document.

Revision History/Date		Corrections	Reviser	Remarks

## 5. Properties

This section introduces EDS SDK APIs for RAW Develop.

For EDS SDK APIs that are not only for RAW Develop, please refer to EDS SDK API document.

Properties of camera and images objects can be retrieved and set by means of **EdsGetPropertyData**, **EdsSetPropertyData**, and other APIs.

If the target object is an image, it has properties besides current settings values—specifically, properties that store settings values at the time the image was shot. Current property settings values are usually indicated, assuming you do not particularly need the previous values. However, by designating a property ID and an OR value for **kEdsPropID\_AtCapture\_Flag** in the arguments for **EdsGetPropertyData**, you can get the properties at the time of shooting. For details, see the description of **kEdsPropID\_AtCapture\_Flag** properties.

For the various properties there are, this section explains the objects they describe and what the properties mean.

### 5.1 Property Lists

Property IDs are listed below. <defined location>EDSDKTypes.h

#### ■ Image Properties

Value	Description
0x00000102	Image orientation
0x00000105	Digital exposure compensation
0x00000106	White balance (light source)
0x00000107	Color temperature setting value
0x00000108	White balance shift compensation
0x0000010d	Color space setting
0x00000114	Picture style
0x00000115	Picture style setting details

#### ■ Develop Properties

Value	Description
0x00000300	Linear processing ON/OFF
0x00000301	Click WB coordinates
0x00000302	WB control value

#### ■ Other

Value	Description
0x0000FFFF	Unknown

Revision History/Date	Corrections	Reviser	Remarks

## 5.2 Property Details

Properties are explained in the following format.

### 5.2.xx PropertyID

The property ID.

#### Description

Explains the role of the property and how to work with it.

#### Target Object

Indicates the "target object" that the property describes and which is subject to operations involving the property.

Properties for which "Access Type" is [Read] can be read by means of objects subject to operations, such as remote cameras. Similarly, an access type of [Write] means the property can be set by means of operations on objects subject to operations.

"Data type number" indicates the enumeration name for data types that can be retrieved by means of **EdsGetPropertySize**.

"Data type" indicates the data type of property data that can be retrieved or set by means of an **EdsVoid** pointer, which is a dummy argument for **EdsGetPropertyData** or **EdsSetPropertyData**.

#### Value

Indicates possible values for the property.

Values are expressed as decimals unless otherwise noted.

#### Note

Considerations when using the property.

Revision History/Date		Corrections	Reviser	Remarks

### 5.2.1 kEdsPropID\_AtCapture\_Flag

#### Description

A supporting property for getting the properties at the time of shooting.  
This property ID cannot be used by itself.

Usually, the properties you can retrieve from objects are the current settings values. However, if the target object is EdsImageRef, when getting image properties, you can get some properties at the time of shooting by designating a property ID and an OR value for **kEdsPropID\_AtCapture\_Flag** in the arguments for **EdsGetPropertyData**.

The property types of values at the time of shooting that can be retrieved are as follows.

Properties that can be retrieved for settings values at the time of shooting
kEdsPropID_DigitalExposure
kEdsPropID_WhiteBalance
kEdsPropID_ColorTemperature
kEdsPropID_WhiteBalanceShift
kEdsPropID_ClickWBPoint
kEdsPropID_WBCoeffs
kEdsPropID_Linear
kEdsPropID_ColorSpace
kEdsPropID_PictureStyle
kEdsPropID_PictureStyleDesc

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read	kEdsDataType_UInt32	EdsUInt32

#### Value

None

Revision History/Date	Corrections	Reviser	Remarks

### 5.2.2 kEdsPropID\_Orientation

#### Description

Indicates image rotation information.

This property can be read or written, regardless of the image compression format (RAW, JPEG, and so on); the access type is Read/Write.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType UInt32	EdsUInt32

#### Value

Value	Description	U: Up D: Down L: Left R: Right
1	The 0th row is at the visual top of the image, and the 0th column is on the visual left-hand side	U L + R D
3	The 0th row is at the visual bottom of the image, and the 0th column is on the visual right-hand side	D R + L U
6	The 0th row is on the visual right-hand side of the image, and the 0th column is at the visual top	L D + U R
8	The 0th row is on the visual left-hand side of the image, and the 0th column is at the visual bottom	R U + D L
Other	Reserved	

#### Note

Rotation information is retrieved from images' Exif information. Thus, images rotated by means of a software tool of computer may be displayed differently from how they would appear using the actual rotation information.

Revision History/Date	Corrections	Reviser	Remarks



### 5.2.3 kEdsPropID\_DigitalExposure

#### Description

Indicates the digital exposure compensation.

As the digital exposure compensation, a value is returned representing the compensation for brightness. This is equivalent to the exposure at the time of shooting as adjusted for the aperture plus or minus several steps.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsType_Rational	EdsRational

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Returns the digital exposure compensation as a kEdsType\_Rational type.

#### Note

- With this property, it is possible to get values at the time of shooting.

Revision History/Date		Corrections	Reviser	Remarks

## 5.2.4 kEdsPropID\_WhiteBalance

### Description

Indicates the white balance type.

### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType_Int32	EdsInt32

### Value

Values defined in Enum EdsWhiteBalance.

Enum EdsWhiteBalance <defined location>EDSDKType.h

Value	Description
0	Auto: Ambience priority
1	Daylight
2	Cloudy
3	Tungsten
4	Fluorescent
5	Flash
6	Manual (set by shooting a white card or paper)
8	Shade
9	Color temperature
10	Custom white balance: PC-1
11	Custom white balance: PC-2
12	Custom white balance: PC-3
15	Manual 2
16	Manual 3
18	Manual 4
19	Manual 5
20	Custom white balance: PC-4
21	Custom white balance: PC-5
23	Auto: White priority

### Note

- If the white balance type is "Color Temperature," to know the actual color temperature you must reference another property (kEdsPropID\_ColorTemperature).
- With this property, it is possible to get values at the time of shooting.

Revision History/Date	Corrections	Reviser	Remarks

### 5.2.5 kEdsPropID\_ColorTemperature

#### Description

Indicates the color temperature setting value. (Units: Kelvin)  
Valid only when the white balance is set to Color Temperature.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType UInt32	EdsUInt32

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

2800–10000, in 100-Kelvin increments.  
5200 represents a color temperature of 5200 K.

#### Note

- To know if the white balance is set to color temperature, refer to another property (kEdsPropID\_WhiteBalance).
- With this property, it is possible to get values at the time of shooting.

### 5.2.6 kEdsPropID\_WhiteBalanceShift

#### Description

Indicates the white balance compensation.

#### Target Object

Target object	Access type	Data type number	Data type
EdsCameraRef	Read/Write	kEdsDataType_Int32_Array	EdsInt32[]
EdsImageRef	Read/Write		

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Array number	Description	Value
0	ValueAB	–9 to +9 0x7FFFFFFF = invalid value Note: 0 means no compensation, (–) means compensation toward blue, and (+) means compensation toward amber.
1	ValueGM	–9 to +9 0x7FFFFFFF = invalid value Note: 0 means no compensation, (–) means compensation toward green, and (+) means compensation toward magenta.

Note: "AB" means compensation toward amber-blue and "GM" toward green-magenta.

#### Note

- With this property, it is possible to get values at the time of shooting.

Revision History/Date	Corrections	Reviser	Remarks

### 5.2.7 kEdsPropID\_ClickWBPoint

#### Description

Indicates the coordinates when an image is clicked to set the white balance.

Only writing is valid.

If you designate coordinates for this property, the white balance value for those coordinates is incorporated in the property kEdsPropID\_WBCoeffs.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Write	kEdsDataType_Point	EdsPoint

#### Value

Designate coordinates within the range of the target image.

#### Note

- With this property, it is possible to get values at the time of shooting.

### 5.2.8 kEdsPropID\_WBCoeffs

#### Description

Indicates the white balance value.

You can apply this value to other image properties, to process images under the same white balance.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType_ByteBlock	EdsInt8[]

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Coefficients to maintain a specific white balance. Use unmodified data from a source image with a white balance you want to copy.

#### Note

- With this property, it is possible to get values at the time of shooting.

Revision History/Date		Corrections	Reviser	Remarks

### 5.2.9 kEdsPropID\_Linear

#### Description

Indicates if linear processing is activated or not.

This property is valid only if 16-bit TIFF or 16-bit RGB has been set for image processing.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType_Bool	EdsBool

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

TRUE: Linear processing  
FALSE: Not linear processing

#### Note

- With this property, it is possible to get values at the time of shooting.

### 5.2.10 kEdsPropID\_ColorSpace

#### Description

Indicates the color space.

If the target object is EdsCameraRef and you designate ColorMatrix in inParam, this property corresponds to the color space setting value of ColorMatrix. Similarly, if you designate the processing parameter in inParam, it indicates that setting value. By using inParam = 0, you can designate the current color space.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType_UInt32	EdsUInt32

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Values of Enum EdsColorSpace.

Enum EdsColorSpace <defined location>EDSDKTypes.h

Value	Description
1	sRGB
2	Adobe RGB
0xFFFFFFFF	Unknown

#### Note

- With this property, it is possible to get values at the time of shooting.

Revision History/Date	Corrections	Reviser	Remarks

### 5.2.11 kEdsPropID\_PictureStyle

#### Description

Indicates the picture style.

This property is valid only for models supporting picture styles.

To get or set the picture style registered in "User Setting," designate user setting 1– (kEdsPictureStyle\_User1– ) in inParam. By using inParam = 0, you can designate the current picture style.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType UInt32	EdsUInt32

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Values defined in Enum EdsPictureStyle.

However, kEdsPictureStyle\_UserX in Enum EdsPictureStyle is not used here.

Enum EdsPictureStyle <defined location>EDSDKTypes.h

Value	Picture style
0x0081	Standard
0x0082	Portrait
0x0083	Landscape
0x0084	Neutral
0x0085	Faithful
0x0086	Monochrome
0x0087	Auto (only for supported models).
0x0088	Fine Detail(only for supported models).
0x0041	Computer Setting 1 (base picture style only)
0x0042	Computer Setting 2 (base picture style only)
0x0043	Computer Setting 3 (base picture style only)

#### Note

- Computer settings (1 and so on) refers to data that was set by designating a picture style file to upload to the camera from a host computer. Computer setting data is registered in the corresponding user setting. (For example, computer setting 1 corresponds to user setting 1). As a user setting, it represents a picture style that users can select.
- Picture styles registered in computer settings always have a base picture style. As for picture styles other than presets, only base picture styles can be retrieved by means of this property value.
- With this property, it is possible to get values at the time of shooting.

Revision History/Date	Corrections	Reviser	Remarks

### 5.2.12 kEdsPropID\_PictureStyleDesc

#### Description

Indicates settings for each picture style.  
This property is valid only for models supporting picture styles.

With **EdsGetPropertyData** or **EdsSetPropertyData**, you can designate a picture style in inParam to set that picture style setting item. By using inParam = 0, you can designate the current picture style.

#### Target Object

Target object	Access type	Data type number	Data type
EdsImageRef	Read/Write	kEdsDataType_PictureStyleDesc	EdsPictureStyleDesc

Note: Write is available as the access type with EdsImageRef only for RAW images.

#### Value

Element	Value	Description
contrast	An integer from -4 to 4	Contrast
sharpness	An integer from 0 to 7	Sharpness Strength
saturation	An integer from -4 to 4	Saturation
colorTone	An integer from -4 to 4	Color tone
filterEffect	0: None 1: Yellow 2: Orange 3: Red 4: Green 0xFFFFFFFF: Unknown	Monochrome filter effect
toningEffect	0: None 1: Sepia 2: Blue 3: Violet 4: Green 0xFFFFFFFF: Unknown	Monochrome tone
sharpFineness	An integer from 1 to 5	Sharpness Fineness
sharpThreshold	An integer from 1 to 5	Sharpness Threshold

#### Note

- Write is available as the access type with EdsImageRef objects only for RAW images. Processed images are read-only.
- With this property, it is possible to get values at the time of shooting.

Revision History/Date		Corrections	Reviser	Remarks

## 6. Appendix

### 6.1 Data Types Used by the APIs

Data types defined under EDSKD are listed in EDSKDTypes.h in C language format.

This section introduces data types unique to EDSKD that are used by EDSKD APIs for RAW Develop.

\* For the most recent type definitions, see the header file EDSKDTypes.h.

\* For data types unique to EDSKD that are used by EDSKD APIs that is not only for RAW Develop, please refer to EDSKD API document.

#### 6.1.1 EdsSaveImageSetting

Use this structure as an argument to EdsSaveImage.

```
typedef struct tagEdsSaveImageSetting{
    EdsUInt32 JPEGQuality;    // 1 (coarse)~10 (fine)
    EdsStreamRef iccProfileStream;
    EdsUInt32 reserved ;
} EdsSaveImageSetting;
```

Revision History/Date	Corrections	Reviser	Remarks