EDSDK API Documentation

Version 3.4.1

Johannes Bildstein

April 18, 2018

Contents

1 Introduction

1.1 New Project Setup

1.1.1 Windows

You will need:

- A .Net coding environment
- .Net 3.5 Framework or higher
- Canon EDSDK DLLs

Steps to set up a new EDSDK API project:

- Open up a .Net coding environment of your choice and create a new .Net project (like a Winforms UI, a WPF UI, a library, a console program, etc.)
- Make sure that you define to compile for specifically either x86 or x64. Depending on which you choose, you also have to provide the correct Canon DLLs.
- Add a reference to the appropriate EDSDKAPI.dll
- Add following EDSDK DLLs to the project and set them to be copied into the output folder. Or of course, you can manually copy them beside the executable
  - DPPDLL.dll
  - DPPLibCom.dll
  - DPPRSC.dll
  - EDSDK.dll
  - EdsImage.dll
  - Mlib.dll
  - Ucs32P.dll
- You're good to go!

1.1.2 Mac OS X

You will need:

- A .Net coding environment
- Mono Framework supporting .Net 3.5 or higher
- Canon EDSDK Framework

Steps to set up a new EDSDK API project:

- Open up a .Net coding environment of your choice and create a new project
- Make sure that you define to compile for specifically either x86 or x64. Depending on which you choose, you also have to provide the correct Canon DLLs.
- Add a reference to the appropriate EDSDKAPI.dll
- Now compile the project so an *.app application bundle is created in the output folder. (do this for all modes (e.g Debug and Release))
- Open the *.app as a directory and create a folder named Frameworks like this: *.app/Contents/Frameworks
- Place EDSDK.framework and DPP.framework into the just created Frameworks folder
- You're good to go!

1.2 Examples

To get you an idea how a project with EDSDKAPI could look like there are a few sample projects. Note that all example projects only target x86 for simplicity.

Console_Net35 A simple console project targeting .Net3.5 that simply connects to a camera, takes a photo, downloads it an closes everything again.

Console_Net45 A simple console project targeting .Net4.5 that simply connects to a camera, takes a photo, downloads it an closes everything again.

WinForms_Basic_Net35 A simple Windows Forms project targeting .Net3.5 that simply connects to a camera and lets the user take a photo and download it.
WinForms_Basic_Net45 A simple Windows Forms project targeting .Net4.5 that simply connects to a camera and lets the user take a photo and download it.
WinForms_Net35 A more advanced Windows Forms project targeting .Net3.5 that shows off some of the most important functions.
WinForms_Net45 A more advanced Windows Forms project targeting .Net4.5 that shows off some of the most important functions.
WPF_Basic_Net35 A simple WPF project targeting .Net3.5 that simply connects to a camera and lets the user take a photo and download it.
WPF_Basic_Net45 A simple WPF project targeting .Net4.5 that simply connects to a camera and lets the user take a photo and download it.
WPF_Net35 A more advanced WPF project targeting .Net3.5 that shows off some of the most important functions.
WPF_Net45 A more advanced WPF project targeting .Net4.5 that shows off some of the most important functions.

1.3 Namespaces

1.3.1 EOSDigital.SDK Namespace

In this namespace, everything related to the SDK is defined. The DLL calls, a few low level methods and SDK enums and classes.

1.3.2 EOSDigital.API Namespace

This is the main namespace you will work with most of the time. It is the actual wrapper around the SDK. It contains classes to handle the SDK, Cameras and RAW-Images, some helper classes and some value storage classes. The content of this namespace will be explained in detail later on.

1.3.3 EOSDigital.Controls Namespace

Here are some base classes and enums for the controls defined

1.3.4 EOSDigital.Controls.WinForms Namespace

This namespace contains controls for Windows Forms projects.

1.3.5 EOSDigital.Controls.WPF Namespace

This namespace contains controls for WPF projects.

2 Class Information

2.1 CanonAPI

This class handles the SDK, connected cameras and you can create RawImage instances. It is the first thing you will need and the last thing to dispose. It is important to keep an instance of this class around as long as you wish to execute commands with the SDK. With the first instance of this class you also initiate the SDK and with the last instance to get disposed, the SDK is also terminated.

2.1.1 Events

CameraAdded

This event fires when a camera connects with the computer. You might want to use the GetCameraList method with this event.

CameraAddedEvent (protected, static)

This is the SDK internal delegate which is just passed over to the public CameraAdded event.

2.1.2 Variables

IsSDKInitialized (static)

States if the SDK is initialized or not

MainThread (protected static)

The main SDK thread where the event loop runs. It's kept alive as long as the SDK is initialized.

2.1.3 Methods

GetCameraList

This method returns a list of connected cameras. If a camera has been connected previously, the same instance of the class is returned.

GetCameraPointerList (protected)

This method returns a list of connected cameras as pointer.

GetRawImage

This method returns a new RawImage instance from a file. This method is in this class because this way it is made sure the SDK is initialized. As a parameter you have to provide the path to the file.

StartConsoleEvents

In a console application, this must be called to get SDK events. For example, if you are waiting for a camera to connect, call this method and in the CameraConnected event, call the StopConsoleEvents method to continue.

StopConsoleEvents

When waiting for SDK events, call this to stop waiting for events and continue the program. You can optionally provide the sleeping time in ms between the EdsGetEvent calls. Default is 200ms.

Dispose

Disposes resources and stops waiting for console events. If called from the last instance, it also terminates the SDK.

Dispose (bool) (protected virtual)

Terminates the SDK and disposes resources depending on the bool value. True is used to also dispose managed object beside unmanaged, false only disposes unmanaged objects.
The usage of this is that from the finalizer/destructor it is called with false and from the Dispose method, it is called with true. This pattern is considered good coding practice.

GetFileThumb (Extension)

This method returns a thumbnail of an image saved on your HD. This image can also be a RAW image. Depending on the extension you use it returns different image objects. This is not available in the base library as there are no imaging classes used.

2.2 Camera

This class represents a physical camera. With it you can open and close a session, set and get various values and communicate with the camera in general. It's the class you will need the most while working with this API.

2.2.1 Events

Most of these events will not be fired from the main thread. This means for frameworks that are not thread-safe you will have to handle it accordingly. This is the case for most UI frameworks like Windows Forms or WPF. Before you modify a control, you might have to invoke on the UI thread.

SDKObjectEvent (protected)

The SDK internal object event. The camera always subscribes to it when a session is opened. It is used to pass the public DownloadReady event if a image has been taken or a video is created. It also passes the public ObjectChanged event.

SDKProgressCallbackEvent (protected)

The SDK internal progress event. The camera always subscribes to it when a session is opened. It is used to pass the public ProgressChanged event.

SDKPropertyEvent (protected)

The SDK internal property event. The camera always subscribes to it when a session is opened. It is used to start the live view download routine and to pass the public PropertyChanged event.

SDKStateEvent (protected)

The SDK internal state event. The camera always subscribes to it when a session is opened. It is used to check if the camera has shut down to release data and pass the public CameraHasShutdown event and also always passes the public StateChanged event.

ProgressChanged

Fires if any process reports progress.

LiveViewUpdated

Fires if the LiveView image is updated

DownloadReady

Fires if an image or video is ready for download. Call one of the DownloadFile methods to get the image or CancelDownload to cancel.

PropertyChanged

Fires if a property has changed

StateChanged

Fires if a state has changed

ObjectChanged

Fires if an object has changed

CameraHasShutdown

This event fires if the camera is disconnected or has shut down

LiveViewStopped

This event fires when the live view loop has ended

SessionStateChanged

This event fires if the cameras session state changes

SessionChanged (static)

This event fires if any cameras session state changes

FileDownloaded (.Net3.5 only)

Fires when a file got downloaded. Called either from DownloadFile or from DownloadFiles (once per file)

FileDeleted (.Net3.5 only)

Fires when a file got deleted. Called for each file passed to DeleteFiles.

FileStreamDownloaded (.Net3.5 only)

Fires when a file is downloaded into a stream

GetAllVolumesFinished (.Net3.5 only)

Fires when a the GetAllVolumes method is finished

GetAllImagesFinished (.Net3.5 only)

Fires when a the GetAllImages method is finished

GetAllEntriesFinished (.Net3.5 only)

Fires when a the GetAllEntries method is finished

ImageDownloaded (.Net3.5 with Extension only)

Fires when an Image is finished downloading from the camera with the DownloadImage method

2.2.2 Variables

Info (protected)

Info about this camera (can be retrieved without an open session)

GlobalIsLiveViewOn (protected static)

States if the live view of any camera is running on the computer or not

MainThread (protected)

Thread for executing SDK commands

IntPtr Reference

Pointer to the SDKs camera object

ID

An ID for the camera object. It's essentially the pointer of the Canon SDK camera object

SessionOpen

States if a session with this camera is open

IsDisposed

States if the camera is disposed. If true, it can't be used anymore

DeviceName

The name of the camera (can be retrieved without an open session)

PortName

The name of the port the camera is connected to (can be retrieved without an open session)

IsLiveViewOn

States if the live view is running on the computer or not

IsLiveViewPaused

States if the live view is paused or not

IsFilming

States the camera is filming or not. This simply checks if the Record property is the value On

IsShutterButtonAvailable

States if SC_PressShutterButton command is available for this camera

IsRecordAvailable

States if Record property is available for this camera

EvfImageInfo

All additional information about a live view image is stored here

KeepOn

If set to true it will override the shutdown timer and keep the camera on. If set to false the camera will switch off at a preset time. This property is set to true by default.

2.2.3 Session and Dispose Methods

Dispose

Releases all data and closes session

Dispose (bool) (protected virtual)

Releases all data depending on the bool value and closes session. True is used to also dispose managed object beside unmanaged, false only disposes unmanaged objects.
The usage of this is that from the finalizer/destructor it is called with false and from the Dispose method, it is called with true. This pattern is considered good coding practice.

OpenSession

Open a new session with camera

CloseSession

Close session with camera

2.2.4 Property Settings List

Note: all of those properties are decorated with a CameraPropertyListAttribute

TvSettingsList Gets the list of possible Tv values for the current camera to set.
AvSettingsList Gets the list of possible Av values for the current camera to set.
ISOSettingsList Gets the list of possible ISO values for the current camera to set.
AEModeSettingsList Gets the list of possible AEMode values for the current camera to set. If the camera does not support software-set values (i.e. it has a hardware dial button), it will return an empty array
MeteringModeSettingsList Gets the list of possible MeteringMode values for the current camera to set.
ExposureCompensationSettingsList Gets the list of possible ExposureCompensation values for the current camera to set.

2.2.5 Properties

Note: all of those properties are decorated with a CameraPropertyAttribute

General

ProductName A string representing the product (i.e. camera) name
OwnerName Indicates a string identifying the owner as registered on the camera
Artist Indicates a string identifying the photographer as registered on the camera
Copyright Indicates a string identifying the copyright information as registered on the camera
MakerName Indicates a string identifying the manufacturer
DateTime Indicates the time and date set on the camera
FirmwareVersion Indicates the camera's firmware version
BatteryLevel Battery state: 0–100% (if IsACPowered is true, this is not valid)
IsACPowered Checks if camera is plugged in (true) or is battery powered (false)
SaveTo Indicates the destination of images after shooting
CurrentStorage Gets the current storage media for the camera
CurrentFolder Gets the current folder for the camera
BatteryQuality Gets the level of degradation of the battery
BodyID Indicates the product serial number
HDDirectoryStructure Gets the directory structure information for USB storage
MyMenu Get or sets the MyMenu items
CFn This is actually implemented as methods because it needs an additional parameter.

Image Properties

ImageQuality Indicates the image quality
FocusInfo Indicates focus information for image data at the time of shooting. This property does not depend on the AF mode at the time of shooting. AF frames in focus are indicated by JustFocus, even during manual shooting. JustFocus values and their meaning:
In focus 1, 16, 32
Focus done but AF still on 17, 33
Out of focus 0, 18, 34
To select a different focus point simply set Selected to 1 and all others to 0. If you want auto selection, set Selected to 1 for all focus points.
WhiteBalance Indicates the white balance type
ColorTemperature Indicates the color temperature setting value. (Units: Kelvin) Valid only when the white balance is set to Color Temperature. Range:2800K - 10000K
Obsolete: Invalid for models which don't support ColorTemperature whitebalance setting.
WhiteBalanceShift Indicates the white balance compensation
ColorSpace Indicates the color space
PictureStyle Indicates the picture style. This property is valid only for models supporting picture styles
ParameterSet Indicates the current processing parameter set on a camera.
Obsolete: Valid only for the EOS 1D Mark II and EOS 1Ds Mark II
PictureStyleDesc Indicates settings for each picture style. This property is valid only for models supporting picture styles

Capture Properties

AEMode Indicates settings values of the camera in shooting mode. When the AE Mode Dial is set to camera user settings, you will get the AE mode which is been registered to the selected camera user setting. Use AEModeSettingsList to get a list of supported values for this camera.
DriveMode Indicates settings values of the camera in drive mode
ISO Indicates ISO sensitivity settings values. Use ISOSettingsList to get a list of supported values for this camera.
MeteringMode Indicates the metering mode. Use MeteringModeSettingsList to get a list of supported values for this camera.
Av Indicates the aperture value. Use AvSettingsList to get a list of supported values for this camera.
Tv Indicates the shutter speed. Use TvSettingsList to get a list of supported values for this camera.
ExposureCompensation Indicates the exposure compensation. Exposure compensation refers to compensation relative to the position of the standard exposure mark (in the center of the exposure gauge). Use ExposureCompensationSettingsList to get a list of supported values for this camera.
FlashCompensation Indicates the flash compensation. Note that flash compensation cannot be retrieved for an external flash.
AFMode Indicates Autofocus mode settings values
AvailableShots Indicates the number of shots available on a camera or host computer
Bracket Indicates the current bracket type. If multiple brackets have been set on the camera, you can get the bracket type as a logical sum.
WhiteBalanceBracket Indicates the white balance bracket amount
LensStatus Returns the camera state of whether the lens attached to the camera. This property can only be retrieved from images shot using models the EOS 50D or EOS 5D Mark II or later
Record The status of the video recording. This property can be used from the EOS 5D Mark II or later and if the camera has video capabilities

Evf Properties

Evf_OutputDevice Indicates the state of the live view output device
Evf_Mode Gets/sets live view function settings
Evf_WhiteBalance Gets/sets the white balance of the live view image. The white balance for the live view image can be set separately from that for the image being shot.

Evf_ColorTemperature Gets/sets the color temperature of the live view image. Just as with the white balance setting for the live view image, the color temperature for the live view image can also be set separately from that for the image being shot. This is applied to the image only when the live view white balance is set to Color temperature.
Evf_DepthOfFieldPreview Turns the depth of field ON/OFF during Preview mode. If Evf_OutputDevice is set to PC and depth of field is being used, the camera will be put in UI Lock status.
Evf_Zoom Sets the zoom ratio for the live view. To get the value use EvfImageInfo
Evf_ZoomPosition Sets the focus and zoom border position for live view. To get the value use EvfImageInfo
Evf_AFMode Set/Get the AF mode for the live view. This property can set/get from the EOS 50D or EOS 5D Mark II or later.

2.2.6 Commands

SC stands for SendCommand. It's added to distinguish from other methods.
Note: these commands are all awaitable in the .Net4.5 version
Note: all of those methods (except the base method) are decorated with a CameraCommandAttribute

SendCommand (protected)

This is the base method that gets called from all the following methods.
command the command that gets sent to the camera
inParam an additional parameter (optional)

SC_TakePicture

Takes a Picture. Consider using the TakePhoto method as this should only be called by specific older models and if not called from a different thread, it could hang the application if no focus is found.

SC_OpenInternalFlash

Opens up the built in flash of the camera.

SC_MovieModeOn

Sets the camera into movie mode. This command allows you to film no matter in which AE mode you are (e.g. Manual or Auto).

SC_MovieModeOff

Switches the movie mode off

SC_ExtendShutDownTimer

Extend camera shutdown time

SC_BulbStart

Start bulb photo. Consider using the TakePhoto method as this could hang the application if no focus is found and is not called from a different thread.

SC_BulbEnd

Stop bulb photo

SC_PressShutterButton

Press cameras shutter button. Consider using the TakePhoto method as this should only be called by specific newer models.
state State of the shutterbutton

SC_DoEvfAf

Do autofocus in live view. This command is only supported by 50D, 5D Mark II or later cameras
state Boolean value to switch it on or off

SC_DriveLensEvf

Change focus of lens in live view
focus Speed and direction of focus movement

SC_DoClickWBEvf

Do whitebalance of live view
x X-coordinate on live view image to do a whitebalance
y Y-coordinate on live view image to do a whitebalance

2.2.7 Status Commands

SSC stands for SendStatusCommand. It's added to distinguish from other methods.
Note: these commands are all awaitable in the .Net4.5 version
Note: all of those methods (except the base method) are decorated with a CameraStatusCommandAttribute

SendStatusCommand (protected)

This is the base method that gets called from all the following methods.
command the status command that gets sent to the camera

SSC_UILock

Lock Camera UI

SSC_UIUnLock

Unlock Camera UI

SSC_EnterDirectTransfer

Enter direct transfer mode

SSC_ExitDirectTransfer

Exit direct transfer mode

2.2.8 Live view

StartLiveView

Starts the live view. It is exactly the same if you set the Evf_OutputDevice property to "PC"

StopLiveView

Stops the live view. With the optional parameter you can specify whether to shut off the live view completely or to just give it back to the camera.
It is exactly the same if you set the Evf_OutputDevice property to "Off" or "Camera" respectively.

PauseLiveView

Pauses the live view. Note that this only applies to the application. The camera is still in live view mode but the frames are not retrieved.

ResumeLiveView

Resumes the live view after pausing

2.2.9 Filming

Note: Filming works only on cameras that have a filming function (which most modern cameras have)
Note: This can only be used if the camera is ready to film. Either call the SC_MovieModeOn method and wait until the Record property has the value "Ready" or set the physical switch on your camera to the movie mode.
Note: the SaveTo property is automatically set to "Camera" and is not set back to the previous value after filming. This is a known issue.

StartFilming

Starts recording a video
PCLiveview If true, the live view will be transferred to the computer otherwise it's shown on the camera

StopFilming

Stops recording a video
saveFilm If true, the DownloadReady event will fire as soon as the video file is created on the camera
stopLiveView If true, the PC live view will stop and will only be shown on the camera

2.2.10 Take Photo

Note: these methods are all awaitable in the .Net4.5 version
Note: these methods all have an overload with a "wait" parameter to execute them synchronously in the .Net3.5 version
Note: The non-Bulb methods either use SC_TakePicture or SC_PressShutterButton depending on availability

TakePhoto

Takes a photo with the current camera settings

TakePhoto (Autofocus)

Takes a photo with the current camera settings with or without autofocus
Note: If the camera does not support PressShutterButton and autofocus is set to false, this method will throw an InvalidOperationException
useAf True to use the autofocus; False otherwise. Note that the lens AF has to be switched on as well to use the autofocus.

TakePhoto (Bulb)

Takes a photo in bulb mode with the current camera settings
Note: The Tv value has to be on Bulb.
delay The delay in seconds before the photo will be taken

TakePhoto (Delay)

Takes a photo with the current camera settings after a delay

TakePhoto (Delay + Autofocus)

Takes a photo with the current camera settings after a delay with or without autofocus
Note: If the camera does not support PressShutterButton and autofocus is set to false, this method will throw an InvalidOperationException
useAf True to use the autofocus; False otherwise. Note that the lens AF has to be switched on as well to use the autofocus.

TakePhoto (Delay + Bulb)

Takes a photo in bulb mode with the current camera settings after a delay
Note: The Tv value has to be on Bulb.
delay The delay in seconds before the photo will be taken
bulbTime The time in ms for how long the shutter will be open

2.2.11 File Handling Methods

Note: these methods are all awaitable in the .Net4.5 version
Note: most of these methods have an overload with a "wait" parameter to execute them synchronously in the .Net3.5 version

FormatVolume

Formats a given camera volume. Get the available volumes with GetAllVolumes.
Warning: All data on this volume will be lost!
volume The volume that will get formatted

GetAllVolumes

Checks for all volumes available on the camera.
Note: Content of volumes is not read
.Net 3.5 Check the GetAllVolumesFinished event for the
.Net 3.5 (Sync) returns an array of CameraFileEntries where each represents a volume
.Net 4.5 returns an array of CameraFileEntries where each represents a volume

GetAllImages

Gets all images (and videos) saved on the cameras memory card(s)
.Net 3.5 Check the GetAllImagesFinished event for the result
.Net 3.5 (Sync) returns an array of CameraFileEntries where each represents an image
.Net 4.5 returns an array of CameraFileEntries where each represents an image

DownloadFiles

Downloads all given files into a folder. To get all images on the camera, use GetAllImages or all files with GetAllEntries
Note: All given CameraFileEntries will be disposed after this. Call GetAllImages or GetAllEntries to get valid entries again.
folderpath The path to the folder where the images will be saved to
images The files that will be downloaded

DeleteFiles

Deletes all given files on the camera. To get all images on the camera, use GetAllImages or all files with GetAllEntries
images The files that will be deleted

GetAllEntries

Gets all volumes, folders and files existing on the camera
.Net 3.5 Check the GetAllEntriesFinished event for the result
.Net 3.5 (Sync) returns a CameraFileEntry with all informations
.Net 4.5 returns a CameraFileEntry with all informations

DownloadFile (to File)

Downloads a file to given directory with the filename in the DownloadInfo. You can change the filename by accessing the DownloadInfo.

Info The DownloadInfo that is provided by the DownloadReady event
directory The directory where the file will be saved to

DownloadFile (to Stream)

Downloads a file into a Stream.

Info The DownloadInfo that is provided by the DownloadReady event
.Net 3.5 Check the FileStreamDownloaded event for the result
.Net 3.5 (Sync) returns a stream containing the file data
.Net 4.5 returns a stream containing the file data

CancelDownload

Cancels the download of an image

Info The DownloadInfo that is provided by the DownloadReady event

DownloadImage (Extension)

This method downloads a jpg image into an image object. Depending on the extension you use it returns different image objects.

.Net 3.5 Check the ImageDownloaded event for the result
.Net 3.5 (Sync) returns an image object
.Net 4.5 returns an image object

2.2.12 Other Methods

Note: these methods are all awaitable in the .Net4.5 version

SetCapacity

Tells the camera how much space is available on the host PC.
This method should be called after opening a session and changing the SaveTo property to either Host or Both, otherwise some cameras think there is not enough space left and throw an error when taking a picture. If you don't know or don't care how much space is left, simply provide a big enough number.

BytesPerSector Bytes per HD sector
NumberOfFreeClusters Number of free clusters on the HD

UILock

Locks or unlocks the camera UI

LockState True to lock, false to unlock

2.2.13 Get Settings (protected)

These are the base methods to get camera settings with various data types.
All of them have these parameters:

propID The property ID
inParam Additional property information (optional)

GetUInt32Setting

Gets a cameras property with the 32 bit unsigned int data type (uint)

GetInt32Setting

Gets a cameras property with the 32 bit signed int data type (int)

GetByteSetting

Gets a cameras property with the 8 bit unsigned int data type (byte)

GetStringSetting

Gets a cameras property with the string data type (string)

GetStructSetting

Gets a cameras property with any SDK struct data type:

- Point
- Size
- Rectangle
- Rational
- Time
- DeviceInfo
- VolumeInfo
- DirectoryItemInfo
- ImageInfo
- SaveImageSetting
- PropertyDesc
- PictureStyleDesc
- FocusPoint
- FocusInfo
- UsersetData
- Capacity

GetInt32ArrSetting

Gets a cameras property with the 32 bit signed int array data type (int[])

GetUInt32ArrSetting

Gets a cameras property with the 32 bit unsigned int array data type (uint[])

GetRationalArrSetting

Gets a cameras property with the Rational struct array data type (Rational[])

GetByteArrSetting

Gets a cameras property with the 8 bit unsigned int array data type (byte[])

2.2.14 Set Settings (protected)

These are the base methods to set camera settings with various data types.
They all have these parameters:

propID The property ID
value The value that will be set
inParam Additional property information (optional)

SetSetting

Sets camera settings with any data type but strings.

SetSetting (string)

Sets string only camera settings. It has this parameter additionally to the others:

MAX The maximum length of the string. 32 characters by default (optional)

SetStructSetting

This method is deprecated. Use SetSetting instead.

2.2.15 Subroutines

SetCapacityBase (protected)

Tells the camera how much space is available on the host PC. This is the unsafe base method that is called safely from the specific frameworks

DownloadEvfMetadata (protected)

Downloads the live view metadata into EvfImageInfo

GetCoordinates (protected)

Converts two ushort values into one uint

SendSDKCommand (protected)

Executes a command to the SDK safely. When the live view is running the command gets queued, otherwise it gets executed immediately

DownloadEvf (protected)

This is the routine that downloads live view images and metadata continuously when the live view is switched on

DownloadData (protected)

This is the base method that downloads any files (e.g. images or videos) from the camera

2.2.16 DownloadToFile (protected)

Downloads any data from the camera to the computer in a file

2.2.17 DownloadToStream (protected)

Downloads any data from the camera to the computer into a stream

GetChildren (protected)

The recursive method to get all folders and files from the camera. It is called from the GetAllEntries method

GetChild (protected)

Gets information about children of a folder in the cameras file system

GetUIntList (protected)

A base method to get an array of camera values. This method is called from the properties listed in the Settings List section

GetHistogram (protected)

A base method to get histogram values. It is called to get the live view histograms

CheckState (protected)

Checks if the camera and SDK state is valid. i.e. if Camera.IsDisposed is false, Camera.SessionOpen is true and CanonAPI.IsSDKInitialized is true.
The SessionOpen property is only checked if the checkSession parameter is true. (it is an optional parameter and defaults to true)

GetBool (protected)

Creates a bool value from an integer. 1 == true, else == false

GetAllVolumesSub (protected) (.Net3.5 only)

Checks for all volumes available on the camera. Note: NOT save to execute while the live view is on! It is called safely from some file handling methods.

2.3 RawImage

This class represents a Canon RAW image. It is possible to retrieve various metadata information about the image and you can process the image and save it as a normal Jpeg or Tiff (8 or 16 bit).

2.3.1 Variables

Reference

Pointer to the raw image

ID

An ID for the raw image object. It's essentially the pointer of the Canon SDK image object

IsDisposed

States if the image has been disposed

FileName

The file name of this raw image

FilePath

The file path of this raw image

2.3.2 Methods

Save (to File)

Converts and saves the image to disk **Note:** in the .Net4.5 version there are async overloads as well
filepath Path to the output image file
targetType Output image type
jpgQuality Quality of saved jpg: 1 = bad quality, small file – 10 = good quality, big file

Save (to Stream)

Converts and saves the image to a stream
outStream Stream to which the image will be saved
targetType Output image type
jpgQuality Quality of saved jpg: 1 = bad quality, small file – 10 = good quality, big file

Dispose

Releases the image object

Dispose (bool) (protected virtual)

Releases the image object depending on the bool value. True is used to also dispose managed object beside unmanaged, False only disposes unmanaged objects.
The usage of this is that from the finalizer/destructor it is called with False and from the Dispose method, it is called with True. This pattern is considered good coding practice.

CheckState (protected)

Checks if the raw image and SDK state is valid. i.e. if RawImage.IsDisposed is false and if CanonAPI.IsSDKInitialized is true.

GetBool (protected)

Creates a bool value from an integer. 1 == true, else == false

2.3.3 Properties

Note: all of those properties are decorated with a RawImagePropertyAttribute
ProductName A string representing the product name of the camera used to shoot the image
OwnerName Indicates a string identifying the owner as registered on the camera used to shoot the image
Artist Indicates a string identifying the photographer as registered on the camera used to shoot the image
Copyright Indicates a string identifying the copyright information as registered on the camera used to shoot the image
MakerName Indicates a string identifying the manufacturer of the camera used to shoot the image
DateTime Indicates the time and date set on the camera used to shoot the image
FirmwareVersion Indicates the firmware version of the camera used to shoot the image
BodyID Indicates the product serial number of the camera used to shoot the image
Orientation Indicates image rotation information
ICCProfile Indicates the ICC profile data embedded in an image
FocusInfo Indicates focus information for image data at the time of shooting
DigitalExposure 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
WhiteBalance Indicates the white balance type
ColorTemperature Indicates the color temperature setting value. (Units: Kelvin)
WhiteBalanceShift Indicates the white balance compensation
Contrast Indicates the contrast. This property is invalid for models supporting picture styles.
–2 to 2 for 20D, Kiss Digital N/350D/REBEL XT, 1D Mark II, 1Ds Mark II
0x7FFFFFFF means Unknown
Obsolete: Use PictureStyleDesc
ColorSaturation Indicates the saturation. This property is invalid for models supporting picture styles. For models supporting picture styles, use PictureStyleDesc
–2 to 2 for 20D, Kiss Digital N/350D/REBEL XT, 1D Mark II, 1Ds Mark II
0x7FFFFFFF means Unknown
Obsolete: Use PictureStyleDesc
ColorTone Indicates the color tone. This property is invalid for models supporting picture styles.
–2 to 2 for 20D, Kiss Digital N/350D/REBEL XT, 1D Mark II, 1Ds Mark II
0x7FFFFFFF means Unknown
Obsolete: Use PictureStyleDesc
Sharpness Indicates the saturation. This property is invalid for models supporting picture styles.
0 to 5 for 1 series models.
-2 to 2 for 20D, Kiss Digital N/350D/REBEL XT
0x7FFFFFFF means Unknown
Obsolete: Use PictureStyleDesc
ColorSpace Indicates the color space

PhotoEffect Indicates the photo effect **Obsolete:** Valid only for the 20D and Kiss Digital N/350D/REBEL XT
FilterEffect Indicates the monochrome filter effect. This property is invalid for models supporting picture styles. For models supporting picture styles, use PictureStyleDesc
Obsolete: Valid only for the 20D and Kiss Digital N/350D/REBEL XT
ToningEffect Indicates the monochrome tone. This property is invalid for models supporting picture styles. For models supporting picture styles, use PictureStyleDesc
Obsolete: Valid only for the 20D and Kiss Digital N/350D/REBEL XT
ColorMatrix Indicates the color matrix.
Obsolete: Valid only for the EOS 1D Mark II and EOS 1Ds Mark II
PictureStyle Indicates the picture style. This property is valid only for models supporting picture styles (modern cameras)
ToneCurve Indicates the tone curve
PictureStyleDesc Indicates settings for each picture style. This property is valid only for models supporting picture styles (modern cameras)
PictureStyleCaption Returns the user-specified picture style caption name at the time of shooting.This property is valid only for models supporting picture styles
Linear 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
ClickWBPoint Indicates the coordinates when an image is clicked to set the white balance. Get the white-balance of the set point with WBCoeffs
WBCoeffs Indicates the white balance value
GPSVersionID Indicates the version of GPSInfoIFD
GPSLatitudeRef Indicates whether the latitude is north or south latitude. The value 'N' indicates north latitude, and 'S' is south latitude
GPSLatitude Indicates the latitude
GPSLongitudeRef Indicates whether the longitude is east or west longitude. 'E' indicates east longitude, and 'W' is west longitude.
GPSLongitude Indicates the longitude
GPSAltitudeRef Indicates the altitude used as the reference altitude. If the reference is sea level and the altitude is above sea level, 0 is given. If the altitude is below sea level, a value of 1 is given and the altitude is indicated as an absolute value in the GPSAltitude. The reference unit is meters.
GPSAltitude Indicates the altitude based on the reference in GPSAltitudeRef. The reference unit is meters.
GPSTimeStamp Indicates the time as UTC (Coordinated Universal Time).
GPSSatellites Indicates the GPS satellites used for measurements
GPSMapDatum Indicates the geodetic survey data used by the GPS receiver
GPSDateStamp A character string recording date information relative to UTC (Coordinated Universal Time). The format is "YYYY:MM:DD."
GPSStatus Indicates the status of the GPS receiver when the image is recorded. 'A' means measurement is in progress, and 'V' means the measurement is Interoperability
AEMode Indicates settings values of the camera in shooting mode
DriveMode Indicates settings values of the camera in drive mode
ISO Indicates ISO sensitivity settings values. The value resembles the ISO sensitivity directly. e.g. a value of 100 means ISO 100
MeteringMode Indicates the metering mode
AFMode Indicates Autofocus mode settings values
Av Indicates the camera's aperture value
Tv Indicates the shutter speed
ExposureCompensation Indicates the exposure compensation.Exposure compensation refers to compensation relative to the position of the standard exposure mark (in the center of the exposure gauge)
FocalLength Indicates the focal length of the lens.When a single-focus lens is used, the same value is returned for the Wide and Tele focal length
Bracket Indicates the current bracket type. If multiple brackets have been set on the camera, you can get the bracket type as a logical sum
WhiteBalanceBracket Indicates the white balance bracket amount

LensName Returns the lens name at the time of shooting
NoiseReduction Indicates noise reduction
RedEye Indicates red-eye reduction
FlashOn Indicates if the flash was on at the time of shooting
FlashMode Indicates the flash type at the time of shooting

2.3.4 Get Settings (protected)

These are the base methods to get image settings with various data types.
All of them have these parameters:
propID The property ID
inParam Additional property information (optional)

GetUInt32Setting

Gets a images property with the 32 bit unsigned int data type (uint)

GetInt32Setting

Gets a images property with the 32 bit signed int data type (int)

GetByteSetting

Gets a images property with the 8 bit unsigned int data type (byte)

GetStringSetting

Gets a images property with the string data type (string)

GetStructSetting

Gets a images property with any SDK struct data type:
- Point
- Size
- Rectangle
- Rational
- Time
- DeviceInfo
- VolumeInfo
- DirectoryItemInfo
- ImageInfo
- SaveImageSetting
- PropertyDesc
- PictureStyleDesc
- FocusPoint
- FocusInfo
- UsersetData
- Capacity

GetInt32ArrSetting

Gets a images property with the 32 bit signed int array data type (int[])

GetUInt32ArrSetting

Gets a images property with the 32 bit unsigned int array data type (uint[])

GetRationalArrSetting

Gets a images property with the Rational struct array data type (Rational[])

GetByteArrSetting

Gets a images property with the 8 bit unsigned int array data type (byte[])

2.3.5 Set Settings (private)

These are the base methods to set image settings with various data types.
They all have this parameters:
propID The property ID
value The value that will be set
inParam Additional property information (optional)

SetSetting

Sets image settings with any data type but strings.

SetSetting (string)

Sets string only image settings. It has this parameter additionally to the others:
MAX The maximum length of the string. 32 characters by default (optional)

SetStructSetting

This method is deprecated. Use SetSetting instead.

2.4 Helper Classes

2.4.1 CameraValue

This class stores a value in three different ways (if possible). The ID as uint, the name as string and the value as double. This class only supports six different properties. These properties are also the ones you can get a list of possible values from the camera. Those properties are:

- Tv
- Av
- ISO
- AEMode
- MeteringMode
- ExposureCompensation

Properties

StringValue The value as a string (use it for display purposes)
UintValue The value as an UInt (ID value for SDK handling)
DoubleValue The value as a double (use it to calculate things)
ValueType The property ID of this value

Additional information

This class implements:

- == operator
- != operator
- Equals
- GetHashCode
- ToString
- implicit conversion to uint

- implicit conversion to string
- implicit conversion to double

2.4.2 DownloadInfo

This class contains information about an object that is ready to download. It usually is used to pass to the DownloadFile method (Camera class). It's also possible to change the filename, check the filesize and if it's a RAW file.

Properties

inRef (protected) Field for the Reference property
dirInfo (protected) Directory item info of the downloadable object
Reference Pointer to the downloadable object
FileName The name of the file. You can change it before you pass it to the DownloadFile method.
Size The files size in bytes
IsRAW States if the file is a RAW file or not

2.4.3 CameraFileEntry

This class represents an entry in the cameras file system. It can contain subentries if it's a volume or folder. If you use one of the API extensions it can contain a thumbnail too (if the file has one).

Properties

Ref (protected) Field for the Reference property
IsDisposed (protected) States if the entry is disposed or not
Reference Pointer to the file entry
Name The name of the entry. (volume name, folder name or file name)
IsFolder States if the entry is a folder or not
IsVolume States if the entry is a volume or not
Entries If the entry is a volume or folder, these are the subentries it contains. It's null if no subentries are present.

Methods

Dispose Releases the entry but not the subentries
Dispose (bool) Releases the entry but not the subentries. True is used to also dispose managed object beside unmanaged, False only disposes unmanaged objects.
DisposeAll Releases the entry and all the subentries (and their subentries and so on)

Additional information

This class implements:
- == operator
- != operator
- Equals
- GetHashCode

2.4.4 STAThread

A class that executes things on an STA thread and provides a method to create an STA thread.

Properties

ExecLock (static readonly) Object that is used for the lock keyword to ensure only one SDK command is executed at a time
IsSTAThread (static) States if the calling thread is in a Single Threaded Apartment or not

IsRunning States if this thread is currently running
ThreadID Managed ID of the associated thread

Methods

Start Starts the execution loop
Shutdown Shuts down the execution loop and waits for it to finish
Invoke (Action) Executes an action on this STA thread. It is ensured that only one action at a time runs. This call is blocking.
Invoke (Func<T>) Executes a function with return value on this STA thread. It is ensured that only one action at a time runs. This call is blocking.
InvokeAsync (Action) (.Net4.5) Same as Invoke(Action) but is awaitable.
InvokeAsync (Func<T>) (.Net4.5) Same as Invoke(Func<T>) but is awaitable.
CreateThread (static) Creates a thread in a Single Threaded Apartment
ExecuteThread (Action) (static) Executes an action on a newly created STA Thread and optionally waits for it to finish executing
ExecuteThread (Func<T>) (static) Executes a function on a newly created STA Thread
ExecuteThreadAsync (Action) (static) (.Net4.5) Same as ExecuteThread(Action) but is awaitable.
ExecuteThreadAsync (Func<T>) (static) (.Net4.5) Same as ExecuteThread(Func<T>) but is awaitable.

2.4.5 Exceptions

CameraException

Derives from System.Exception and additionally stores the SDK error name.

CameraSessionException

Derives from System.Exception and states a problem with the session state of the camera

SDKStateException

Derives from System.Exception and states a problem with the state of the Canon SDK

ExecutionException

Derives from System.Exception and is thrown when an error happened while executing something on an STA thread

2.4.6 SDKErrorHandler

This static class handles SDK errors and provides events for errors that happened on non-main threads. If you don't subscribe to the events, exceptions will be thrown in the threads and may crash the program.
There are a few errors that are considered non-severe who don't interrupt the code. They can be removed and added with the NonSevereErrors property but a few are defined per default already: (values are from the ErrorCode enum):
TAKE_PICTURE_AF_NG Autofocus failed
TAKE_PICTURE_CARD_NG Error at writing to memory card
TAKE_PICTURE_CARD_PROTECT_NG Memory card is write protected
TAKE_PICTURE_LV_REL_PROHIBIT_MODE_NG *No documentation*
TAKE_PICTURE_MIRROR_UP_NG Image could not be taken because mirror is up
TAKE_PICTURE_MOVIE_CROP_NG *No documentation*
TAKE_PICTURE_NO_CARD_NG No memory card is in the camera
TAKE_PICTURE_NO_LENS_NG No lens is attached to the camera
TAKE_PICTURE_SENSOR_CLEANING_NG Camera is busy cleaning the sensor
TAKE_PICTURE_SILENCE_NG Camera is busy performing silent operations
TAKE_PICTURE_SPECIAL_MOVIE_MODE_NG *No documentation*
TAKE_PICTURE_STROBO_CHARGE_NG Strobe is currently charging and not ready

LENS_COVER_CLOSE Lens cover is closed
DEVICE_BUSY Camera is currently busy

Methods

CheckError (ErrorCode) This method checks for an error in SDK calls and checks how to treat it
CheckError (uint) This method checks for an error in CanonSDK.EdsRelease and CanonSDK.EdsRetain calls. If no error is found, the number of references is returned.
ReportError Reports an error that happened in a threading environment and passes it to the SevereErrorHappened event

2.4.7 SDKStream

A Stream encapsulating an unmanaged SDK Stream. This class can be used to overcome the differences between SDK versions. With the various constructor overloads it can be a file or memory stream and can also be created from an already existing SDK stream.

Properties

CanRead Gets a value indicating whether the current stream supports reading. (Currently always returns true)
CanSeek Gets a value indicating whether the current stream supports seeking. (Currently always returns true)
CanWrite Gets a value indicating whether the current stream supports writing. (Currently always returns true)
Length Gets the length in bytes of the stream.
Position Gets or sets the position within the current stream.
Reference Pointer to the underlying SDK stream

Methods

Flush Clears all buffers for this stream and causes any buffered data to be written to the underlying device. This is not applicable to the SDK and therefore does nothing.
Read Reads a sequence of bytes from the current stream and advances the position within the stream by the number of bytes read.
Seek Sets the position within the current stream. Same as setting the Position property.
SetLength Always throws a NotSupportedException
Write Writes a sequence of bytes to the current stream and advances the current position within this stream by the number of bytes written.

2.4.8 EvfMetadata

This is a container for all metadata of a live view image and you can set which values should be read (for performance reasons).

Properties

CoordinateSystem The coordinate system of the live view image
IsCoordinateSystemSet No metadata, it simply states if the CoordinateSystem field has been set.
Zoom The zoom ratio of the live view
HistogramStatus The display status of the histogram
ZoomRect The focus and zoom border rectangle for live view
ZoomPosition The focus and zoom border position of the live view
ImagePosition The cropping position of the enlarged live view image
HistogramY The brightness histogram
HistogramR The red histogram
HistogramG The green histogram

HistogramB The blue histogram

2.4.9 CFn

Custom functions are detailed settings for the cameras behavior. Note that the supported values vary between camera models.

CFnCollection

A Collection of all known custom functions. (This list is likely incomplete because there is no documentation about it)

Methods

GetCFnDescription (int) Gets the custom function description of a given ID
GetCFnDescription (CFnSelection) Gets the custom function description of a given selection
GetCFnSelection Gets the custom function selection of a given type ID and value ID
TryGetCFnSelection Safely gets the custom function selection of a given type ID and value ID

Structs

There are a few structs to store the IDs and descriptions in a convenient way.
They all implement:

- == operator
- != operator
- Equals
- GetHashCode

CFnSelection

Custom function selection
Type Type of selection
CFnValue Value of the selection

CFnDescription

Custom function description
Type Custom function type
Values Possible values of this type
Methods:
Select Selects one of the values and returns it as CFnSelection
Find Finds one of the values by the ID
TryFind Safely finds one of the values by the ID

CFnValue

Custom function value
ID ID of this value
Description Description of this value

2.4.10 ValueBase

This is the base class for classes storing all possible values (regardless of the camera) for the same properties as the CameraValue class supports. It provides methods to get a CameraValue from an uint, string or double. These classes are here because it is difficult to juggle with the different possibilities of values. An uint is needed for communication with the camera, a string to show in the UI and a double to calculate.

Subclasses

AvValues for aperture values
TvValues for shutter speed values
ISOValues for ISO values
ExpCompValues for exposure compensation values
AEModeValues for shooting mode values
MeteringModeValues for metering mode values

Methods (static)

These methods are implemented in all subclasses.
GetValue (uint) Get the value from an uint out of all possible values. It has to be an exact match, otherwise an exception is thrown.
GetValue (string) Get the value from a string out of all possible values. It has to be an exact match, otherwise an exception is thrown.
GetValue (double) Get the value from a double out of all possible values. It searches for the closest representation and therefore might not return the exact input value.
GetValue (uint + List) Get the value from an uint out of given possible values. It has to be an exact match, otherwise an exception is thrown.
GetValue (string + List) Get the value from a string out of given possible values. It has to be an exact match, otherwise an exception is thrown.
GetValue (double + List) Get the value from a double out of given possible values. It searches for the closest representation and therefore might not return the exact input value.

2.4.11 WrapStream

A System.IO.Stream encapsulating another System.IO.Stream created by the SDK. Its intention is to fix a bug in WPF with a memory leak when creating images from a stream.
It is only available in the WPF specific binaries.

2.4.12 CFUrl

Wrapper class for the CFUrl object on Mac OS X. It is needed for the CanonSDK.EdsCreateFileStreamEx method.

2.4.13 Helper Structs

They all implement:
- == operator
- != operator
- Equals
- GetHashCode

WhiteBalanceShift

Simple container for the WhiteBalanceShift property (Camera and RawImage class)

WhiteBalanceBracket

Simple container for the WhiteBalanceBracket property (Camera and RawImage class)

FlashMode

Simple container for the FlashMode property (RawImage class)

GPSPosition

Simple container for the GPSPosition property (RawImage class)

GPSTimeStamp

Simple container for the GPSTimeStamp property (RawImage class)

FocalLengthInfo

Simple container for the FocalLength property (RawImage class)

2.5 Attributes

There are a number of attributes that decorate properties and command methods. They can help to find and execute those properties/methods via reflection.

CameraPropertyAttribute Marks a property as Camera property
CameraPropertyListAttribute Marks a property as Camera property list
RawImagePropertyAttribute Marks a property as RawImage property
CameraCommandAttribute Marks a method as CameraCommand
CameraStatusCommandAttribute Marks a method as CameraStatusCommand

2.6 Delegates

For all of the events there are specific delegates defined.
ProgressHandler A delegate for progress
LiveViewUpdate A delegate to pass a stream
DownloadHandler A delegate to report an available download
PropertyChangeHandler A delegate for property changes
StateChangeHandler A delegate for camera state changes
ObjectChangeHandler A delegate for property changes
StringUpdate A delegate for simple string updates
StreamUpdate A delegate for simple stream updates
CameraAddedHandler A delegate to inform of an added camera
CameraSessionHandler A delegate to inform of a session state of a camera
SDKExceptionHandler A delegate to inform of SDK exceptions
GeneralExceptionHandler A delegate to inform of exceptions
ImageEventHandler (System.Drawing) A delegate to handle events with bitmap parameter
ImageSourceEventHandler (WPF) A delegate to handle events with ImageSource parameter

2.7 SDK Structs

There are a number of structs that are used to communicate with the SDK.
They all implement:

- == operator
- != operator
- Equals
- GetHashCode

2.7.1 Point

A point with X and Y coordinates.
Also implements:

- ToString: "X;Y"
- implicit conversion to System.Drawing.Point (only with the System.Drawing extension)
- implicit conversion from System.Drawing.Point (only with the System.Drawing extension)

2.7.2 Size

A struct with Width and Height values.
Also implements:

- ToString: "Width;Height"
- implicit conversion to System.Drawing.Size (only with the System.Drawing extension)
- implicit conversion from System.Drawing.Size (only with the System.Drawing extension)

2.7.3 Rectangle

A rectangle with X and Y coordinates and Width and Height values.
Also implements:

- ToString: "X;Y;Width;Height"
- implicit conversion to System.Drawing.Rectangle (only with the System.Drawing extension)
- implicit conversion from System.Drawing.Rectangle (only with the System.Drawing extension)

2.7.4 Rational

A rational number with Numerator and Denominator.
Also implements:

- ToString: "Numerator/Denominator"
- implicit conversion to double
- implicit conversion to decimal

2.7.5 Time

A struct storing time information from Year to Milliseconds.
Also implements:

- ToString: same as DateTime.ToString()
- implicit conversion to System.DateTime
- implicit conversion from System.DateTime

2.7.6 DeviceInfo

Stores info about a connected camera device.

2.7.7 VolumeInfo

Stores info about a camera volume.

2.7.8 DirectoryItemInfo

Stores info about an Directory Item like a folder or a file.

2.7.9 ImageInfo

Stores info about an image.

2.7.10 SaveImageSetting

Stores info about how an image will be saved.

2.7.11 PropertyDesc

Stores info about a Property Description.

2.7.12 PictureStyleDesc

Stores info about a Picture Style Description.

2.7.13 FocusPoint

Stores info about a single focus point.

2.7.14 FocusInfo

Stores info about focus.

2.7.15 UsersetData

Stores info about User WhiteBalance (PC set1,2,3); User ToneCurve; User PictureStyle dataset.

2.7.16 Capacity

Stores capacity info of a data storage.

2.7.17 MyMenuItems

Stores MyMenu items.

3 Controls

Note: WPF controls should be manually disposed when going out of scope (e.g. when closing a window) by calling the Dispose method. If this is not done there could be performance issues because it would still continue to read the live view data even if it's not displayed anymore.
Windows Forms controls are automatically disposed.

3.1 LiveView

Displays the live view of a camera and allows user interaction to focus or set the zoom/focus rectangle. Depending on what the FocusMode property is set, the control behaves differently:

Off Nothing happens. i.e. interaction is ignored
SetPosition When clicking on the control, the Camera.Evf_ZoomPosition is set to the click position but no focusing is done
Hold While the input (e.g. mouse) is pressed, focusing is done at that position
Tap After clicking on the control, focusing is done at that position for the time specified with the FocusTimeout property

3.1.1 Properties

MainCamera The camera of which the live view is displayed. If no camera is defined, it will assign the first camera that opens a session.
ImagePosition Alignment of the live view image. You can combine them with a logical OR.
ImageTransform Transformation of the live view image. You can combine them with a logical OR.
EnableZoomRectangle States if the live view zoom rectangle gets drawn
EnableFocusRectangles States if the live view focus rectangles get drawn
AlloFocusPointSelection States if a focus point can be selected by clicking on it on the control
FocusMode Live view focus mode on interaction with the control
FocusTimeout Focus time in milliseconds when FocusMode is set to FocusMode.Tap
FocusResultTime Time in milliseconds to show the result of the autofocus
FocusActive States if the focus is currently active

ZoomRectangleColor Color of the zoom rectangle
SelectedFocusRectangleColor Color of a selected focus rectangle
UnselectedFocusRectangleColor Color of an unselected focus rectangle
InFocusRectangleColor Color of a focus rectangle that is in focus
OutOfFocusRectangleColor Color of a focus rectangle that is out of focus
FocusHoverRectangleColor Color of a focus rectangle over which the mouse is hovered

3.1.2 Methods

GrabFrame Grabs the current live view frame (i.e. the last frame that got displayed)

3.1.3 Extending Control Drawing

For custom overlays you can inherit from the LiveView control class and override these methods:

CustomFilter

Gets called when the live view is on and before the frame is transformed. It passes the current live view frame as a parameter which can be analysed and/or manipulated. The WPF version also needs to return an image again. If you don't modify it, just return the same image, otherwise return the new one. If you have changed the size of the frame, you must set the IsSizeSet field to false to ensure the frame is drawn at the right size and position.

CustomFilter

Gets called when the live view is on and after the frame has been painted but before the zoom rectangle is painted. The first parameter is of type PaintEventArgs for Windows Forms and DrawingContext for WPF. They can be used to paint any custom graphics you like. The second parameter is of type TransformationInfo. This class holds information about the currently drawn frame:
Left Distance between control X:0 and image X:0 where (0,0) is the top left corner
Top Distance between control Y:0 and image Y:0 where (0,0) is the top left corner
Width Drawing width of the image
Height Drawing height of the image
Rotation Rotation of the image
MirrorHorizontal States if the image is mirrored horizontally or not
MirrorVertical States if the image is mirrored vertically or not
HasTransformation States if the image is transformed (rotated or mirrored)
HasRotation States if the image is rotated
IsVertical States if the image is rotated to a vertical position
HasMirror States if the image is mirrored

Note that Width and Height is not the size of the original live view frame (i.e. what the camera sends) but the size it's drawn within the control. Also, Width is always the size in control-X (i.e. left/right) and Height the size in control-Y (i.e. up/down) even if the frame is rotated vertically.

CustomInactivePaint

Gets called when the live view is off and the background has been cleared with the Background color property. The only parameter is of type PaintEventArgs for Windows Forms and DrawingContext for WPF. They can be used to paint any custom graphics you like.

3.2 LiveViewHistogram

Displays the histogram of the live view image. Note that you have to turn on the histogram in the camera. For example, to get the brightness histogram: Camera.EvfImageInfo.GetHistogramY = true;

3.2.1 Properties

MainCamera The camera of which the live view histogram is displayed. If no camera is defined, it will assign the first camera that opens a session.
ShownHistogram The channel of the histogram that is shown.
Forecolor Defines the color in which the histogram will be drawn.

4 Appendix

4.1 Known Issues

4.1.1 Thumbnails for CameraFileEntry in SDK 3.x

Getting the thumbnail image for the CameraFileEntry class is broken in the SDK 3.x versions. The error code is always FILE_FORMAT_UNRECOGNIZED and the image data actually seems to be corrupted. I suspect that there is a bug in the EdsDownloadThumbnail method.

4.1.2 Mac OS X and Threading

On Mac OS X the SDK always uses the main thread of the program regardless on which thread EdsInitialize is called. This means you need to use new CanonAPI(true) to create a new CanonAPI instance and subsequently all synchronous calls to methods from the Camera class must be executed on a different thread or you will create a deadlock.

4.1.3 Artist and Copyright property

This is only confirmed with the 40D (a 7D shows no such behavior):
The Artist and Copyright camera property give back an INTERNAL_ERROR if you get them. If the property is set first and then get, it works, but only until the camera is connected again. It seems to be an SDK error since other string properties work fine and it returns an internal error.

4.1.4 Random State Events

This is only confirmed with the 40D (a 7D shows no such behavior):
The StateEvent is randomly fired with the last bulb time. Usually after setting properties.

4.2 Not Implemented

This is a list of known things that are not implemented for various reasons.

4.2.1 Camera

- **Properties**
  Evf_FocusAid There is no documentation about how this property is defined.
  Evf_ImageClipRect There is no documentation about how this property is defined.
  DepthOfField There is no documentation about how this property is defined.
  EFCompensation There is no documentation about how this property is defined.
  JpegQuality Not sure how to handle it. It is only valid for the EOS 1 series.
- **Property IDs**
  16778275 This property changes before and after filming. There is no documentation about it.
  16777239 This property changes before and after filming. There is no documentation about it.
- **InParam** The property is implemented, but inParam is not.
  HDDirectoryStructure Not sure how to handle it.
  ColorSpace Not sure how to handle it.
  PictureStyle Not sure how to handle it.
  PictureStyleDesc Not sure how to handle it.

4.2.2 RawImage

- **Properties**
  JpegQuality Not sure how to handle it. It is only valid for the EOS 1 series.

Sharpness The 1D and 1Ds use an Int32 array instead of an Int32. Since this property is obsolete already (only works with cameras that don't support picture styles), it has not been implemented yet. The normal Sharpness property with Int32 is implemented though.

- **InParam** The property is implemented, but inParam is not.
  ColorSpace Not sure how to handle it.
  PictureStyle Not sure how to handle it.
  PictureStyleDesc Not sure how to handle it.

4.3 Changelog

Version 3.4.1

- Compatibility with EDSDK version 3.8

Version 3.4.0

- Enable setting the AFMode property (only possible when the live view is off)
- Enable setting the FocusInfo property
- Rework the filming logic to be more robust
- Add command to set the camera into movie mode from any other AE mode (e.g. from Manual or Auto)
- Add command to open the internal flash
- Add Evf_TimeStamp property ID
- Add focus point handling to live view control
- Add FocusPoints property to EvfMetadata class
- Improve detecting when the autofocus has finished in the live view control
- Example projects were missing x64 build targets

Version 3.3.9

- Fix exception when having more than one subscriber to the error events
- Add CustomFilter method to the live view controls to analyse and/or manipulate the live view frame directly

Version 3.3.8

- Add CustomFramePaint and CustomInactivePaint methods to the live view controls to create custom overlays

Version 3.3.7

- Implement focusing functionality for the live view controls

Version 3.3.6

- Fix error when setting Evf properties like Evf_ZoomPosition

Version 3.3.5

- Added TakePhoto overloads to define if autofocus should be used or not

Version 3.3.4

- Fix exception when having multiple WPF live view controls
- Fix exception when disposing CanonAPI without disposing Camera before and the live view is on
- Fix bug where only the first instance of CanonAPI would fire the CameraAdded event

Version 3.3.3

- Fix exception when disposing camera with invalid reference pointer because it shut down with a closed session
- Fix incorrect rendering of WinForms live view when transforming image
- The GrabFrame method of the live view controls now returns null when there is no frame (i.e. live view is not running)
- Improved live view controls in safety and correctness
- Added live view image rotation for 90, 180 and 270 degrees in ImageTransformation enum
- Ensure that the shutter button is set to off if an error happens while taking a photo

Version 3.3.2

- Fix possible deadlock when updating the UI with the LiveViewStopped event after calling Camera.Dispose with the live view on
- Fix possible error in SDK events that are fired after the Camera is already disposed
- Improve responsiveness of command execution

Version 3.3.1

- CanonSDK.InitializeVersion could cause a PInvokeStackImbalanceException (only in Debug mode and if MDA was enabled in Visual Studio)
- Ensure that all Camera objects are disposed in all cases
- Sometimes the TakePhoto method took more than one photo when the ShutterButton command is available
- Camera.DownloadToStream did not reset stream position to zero after downloading the image
- Added LiveViewStopped event that fires when the live view loop exits
- The live view did not start when in recording mode but not actively recording
- The WPF live view control could throw an exception when calling Camera.Dispose while the live view was running
- Reduced memory footprint of WinForms live view control
- Live view and histogram controls now clear the view once the live view has stopped
- Live view controls now refresh when a property has changed
- Controls now have a PropertyChanged method
- Fix typo of white balance name in the example UIs
- Fix error of using disposed camera in example UIs
- Smaller touch-ups in the example UIs

Version 3.3.0

- Added several new AEModeValues
- Added Enum value EvfOutputDevice.Filming
- Added Enum value StorageType.CFast
- Added Enum value PictureStyle.FineDetail
- Added Enum value Error.PTP_DEVICE_BUSY
- Several fixes for using the live view while filming
- CanonSDK method EdsCreateStreamEx is actually called EdsCreateFileStreamEx and uses CharSet.Unicode
- Added comments to FileCreateDisposition Enum and renamed TruncateExsisting to TruncateExisting
- Made RawImage reading and saving more secure and added Unicode file path support
- SetStructSetting now calls SetSetting internally and is marked Obsolete
- Added overload to CanonAPI to use calling thread as main SDK thread instead of new dedicated thread
- Added ErrorHandler.CheckError overload that only checks for ErrorCode.OK
- Added IsRecordAvailable property to Camera class that states if the Record property is available for this camera
- SetCapacity switched BytesPerSector and NumberOfFreeClusters parameters
- Added CFUrl class to support the EdsCreateFileStreamEx method on Mac OS X
- STAThread created underlying thread twice; once in the constructor and again in the Start method

- Support for SDK version 3.4
- Added 64bit methods to CanonSDK class for SDK version >=3.4
- Added Size64 to DownloadInfo which is the file size as (unsigned) int64
- Added SharpFineness and SharpThreshold to PictureStyleDesc for SDK version >=3.2
- Added Size64 to DirectoryItemInfo for SDK version >=3.4
- Added GetPropertyData method for PictureStyleDesc in CanonSDK class that works for any SDK version. (the PictureStyleDesc struct changed in SDK v3.2)
- Generic GetPropertyData method in CanonSDK works now for any SDK version.
- Added InitializeVersion method to CanonSDK class that checks for significant SDK versions
- Added IsVerGE34 property to CanonSDK class that states if the used SDK version is >=3.4
- SDKStream class is now using only SDK methods internally and is safe to use regardless of the SDK version
- Smaller safety and performance improvements
- CLS Version:
  - Evf_OutputDevice could return invalid value because of SDK bug

Version 3.2.2

- When the last instance of the CanonAPI class is disposed with the Dispose method, all connected Cameras are disposed as well
- Smaller safety improvements

Version 3.2.1

- When waiting for work to finish on the STA thread, it had a 1ms wait timeout. It's no error but more work than necessary
- Added ISO values: 16000 and 20000
- Added KeepOn property to override the shutdown timer of the camera
- The Camera properties Artist and Copyright were limited to 32 characters when they should be limited to 64
- Smaller improvements of the WPF live view control
- Setting Evf_OutputDevice to PC while the live view was on started another live view thread
- CLS Version:
  - Getting a list of possible camera values returned wrong values because of a wrong uint cast

Version 3.2.0

- Fixed Xaml error in WPF controls.
- A CLS compliant version of the library is now available additionally. If it's working reliably, this will replace the current version in the long run
  - It has a few minor breaking changes compared to the normal version:
  - TakePhoto with bulb parameters is now called TakePhotoBulb
  - All methods, structs, enums and classes that used unsigned values now used signed values. Mostly from uint to int
  - Some methods to get values from the SDK still use unsigned types for completeness. They are marked with CLSCompliant(false)

Version 3.1.6

- Threading model has been refactored. API has a dedicated event loop now and every camera instance has its own execution thread. This makes all SDK calls much more deadlock safe
- All API events (from Camera, CanonAPI and ErrorHandler class) are now fired on a new thread. It was mostly like this before but now it's ensured
- STAThread refactoring. It's not static anymore and methods are now called Invoke instead of Execute-Safely
- All examples have been adapted to the mentioned changes
- The AvValues class has now static fields for the Auto and Invalid values

- The TvValues class has now static fields for the Auto, Bulb and Invalid values
- The ISOValues class has now static fields for the Auto and Invalid values
- The ExpCompValues class has now static fields for the Zero and Invalid values

Version 3.1.5

- The .Net3.5 version now has the option to execute asynchronous methods synchronous (by providing an additional "wait" parameter or an additional ...Sync method)
- Added ExecuteThreadSafely methods to the STAThread class to execute something on a newly created thread

Version 3.1.4

- Checking for SC_PressShutterButton availability at OpenSession could make some cameras unusable (either hanging or DEVICE_BUSY error)

Version 3.1.3

- The Camera.DownloadFile(DownloadInfo) method for .Net3.5 now disposes the stream after firing the event
- The SDKStream class was broken because it took the wrong parameter. It now has two constructors that takes either both object and stream or only object
- The GetPropertyData string was broken

Version 3.1.2

- The System.Drawing CanonAPI.GetImage method returned an invalid Bitamp. Mainly affected was the CameraFileEntry.Thumbnail property
- Smaller security updates

Version 3.1.1

- Added MyMenu Camera property. Note that probably not all enum values are defined yet since they are not documented
- Added MyMenuItems struct for MyMenu property
- Added MyMenuID enum for MyMenu property
- Added CFn Camera property. Note that probably not all values are defined yet since they are not documented
- Added GetPropertyData method for all Canon SDK data types as well as a generic one
- Added GetSetting method for all Canon SDK types in Camera and RawImage classes
- Added GrabFrame method to the live view control
- Added ImageTransform property to live view control
- Fixed a missing closing event in the Windows Forms .Net4.5 example which caused it to not close properly

Version 3.1.0

- Added EDSDKAPI.Controls.WinForms assembly with Windows Forms controls
- Added EDSDKAPI.Controls.WPF assembly with WPF controls
- Added LiveView control for both WPF and WinForms
- Added LiveViewHistogram control for both WPF and WinForms
- Added x64 assemblies to all projects (except for the examples)
- Added SessionStateChanged and SessionChanged events to the Camera class
- Added FileDeleted event to .Net3.5 Camera
- Added a IsSDKInitialized property to the CanonAPI class
- Added a IsShutterButtonAvailable property to the Camera class to check if the SC_PressShutterButton command is available for the camera

- Added an ID property to all attributes that defines the SDKs PropertyID, CameraCommand ID or CameraStatusCommand ID
- Added a few more example projects: Console .Net3.5/.Net4.5 and basic UI examples for WPF and WinForms .Net3.5/.Net4.5
- Added CameraSessionException and SDKStateException to handle specific Exceptions better
- Every critical SDK call is now properly executed on an STA thread. This resolves some of the freezing issues.
- Fixed a bug where already disposed camera objects would be returned from the GetCameraList method in the CanonAPI class
- Methods in the Camera class now properly throw exceptions when called and the class is disposed
- If the SDK is terminated and a Camera or RawImage method is called, an SDKStateException is thrown
- The UI examples with live view use the LiveView control now
- The TakePhoto methods now use either SC_TakePicture or SC_PressShutterButton depending on availability
- The live view download routine enforces a context switch now. Without that, the live view could block events on single threaded machines in some cases
- Smaller security improvements and bug fixes
- Reviewed the code documentation as well as this document

Version 3.0.4

- ValueBase.GetValue with string and int overload return a CameraValue with value "invalid" instead of throwing an exception. If an invalid value is not available, it still throws an exception.
- AvValues and TvValues have now an "Auto" value for when the camera is in an automatic mode (only applies to newer cameras)
- ExpCompValues include now values from ± 5 to $\pm 3\ 1/3$

Version 3.0.3

- The Cameras already in use would not be stored correctly.
- The DownloadToStream method didn't return a valid stream.

Version 3.0.2

- When having more than one instance of the Camera class with the same physical camera (i.e. with the same pointer). The events would not fire on any instance when one of those got disposed (by hand or by garbage collection)
- The CanonAPI class could only have one instance and threw an exception after that. Now multiple instances can be created.
- Added a new DownloadFile method that downloads data into a stream instead of a file.
- Outsourced the logic of both DownloadFile methods into protected methods (not live view safe): DownloadToFile and DownloadToStream
- CanonAPI.GetCameraList now returns the same instance of the Camera class as from a previous call. So now there can't be two Camera instances with the same SDK pointer.

Version 3.0.1

- SC_DoEvfAF had a wrong parameter type (EvfAFMode). It has been replaced with bool (for AF on or off).
- The GetImage method (and with that getting the file thumbnail) in the WPF extension was broken.

Version 3.0.0

- Added .Net4.5 specific version of the API with async/await
- Added example projects for WPF and WinForms in .Net3.5 and .Net4.5 each
- Added multi-camera support (beta: yet to be thoroughly tested (especially with more than two cameras))
- Added implicit operators to some SDK structs and API classes (e.g. SDK.Time to System.DateTime)
- All SDK and API structs are now comparable (with $==$, $!=$, Equals and GetHashCode)

- Some API classes (where it makes sense) are now comparable (with ==, !=, Equals and GetHashCode)
- Dispose methods and Finalizer/Destructor are now implemented in a good coding standard
- Some classes and methods have been renamed to fit its purpose better (some names were misleading or wrong after the updates)
- In .Net3.5 the file handling methods in the camera class have no return value anymore but use events now. They are now safe to use while the live view is on
- Complete overhaul of the exception handling inside the API
- Reworked parts of the code documentation and added exception tags
- Formerly sealed classes have been opened and private variables and methods have been made protected where appropriate
- The pointer to the underlying SDK object has been made public (read-only) in some classes
- The MonoMac (Cocoa) extension has been removed. MonoMac seems to be a dead project. Mac OSX in general is still supported though
- Various smaller security and bug fixes