

IPEVO Camera API for Windows C# v1.0.0.1

1. Steps of implementation

1-1. Add library files AitUVCExtApi.dll, CameraKit.dll, DirectShowLib-2005.dll, Newtonsoft.Json.dll, UsbLibrary.dll to your project.

1-2. Use library in your code

```
using com.ipevo.Presenter.CameraKit;  
using com.ipevo.Presenter.Toolkit;
```

2. Find the IPEVO Camera List

```
CameraManager cameraManager = CameraManager.sharedManager;  
cameraManager.startMonitor();  
this.cameraListComboBox.ItemsSource = cameraManager.cameras;  
  
cameraManager.stopMonitor();           //"must" call this before your Application Exit
```

3. Camera Capability

Ex:

```
bool wb_capability = selectedCamera.hasCapability(Camera.Capability.WhiteBalance);
```

All capability list

Camera.Capability.Exposure	Exposure
Camera.Capability.AutoExposure	AutoExposure, Exposure Lock
Camera.Capability.WhiteBalance	White Balance value
Camera.Capability.ManualFocus	Manual Focus value

4. Listen Notification

Ex:

```
NotificationCenter.defaultCenter.addNotificationObserver(CameraManager.Notification.HID_S_AF, new NotificationCenter.NotificationObserver(camera_focus_mode));
```

Notification Message List:

```
CameraManager.Notification.DeviceDetached  
CameraManager.Notification.DeviceAttached  
CameraManager.Notification.HID_SNAP_SHOT_BTN  
CameraManager.Notification.HID_FOCUS_BEGIN  
CameraManager.Notification.HID_FOCUS_FINISH  
CameraManager.Notification.HID_S_AF  
CameraManager.Notification.HID_C_AF  
CameraManager.Notification.CAMERA_EXPOSURE_CHANGE
```

Listening Notification allows you to get notified when camera status changes.

Ex:

```
private void camera_focus_mode(string str, object sender, object userInfo)  
{  
    //parameter1, sender from  
    //parameter2, userInfo is hashtable  
    // userInfo["Camera"]: type Camera  
    // userInfo["FocusMode"]: type int, 0 S-AF, 1 C-AF  
  
    if ((int)((userInfo as Hashtable)["FocusMode"]) == 0)  
    {
```

```

        Console.WriteLine(((userInfo as Hashtable)["Camera"] as
Camera).CameraInstanceName.ToString() + " " +
                        "S-AF");
    }
    else if ((int)((userInfo as Hashtable)["FocusMode"]) == 1)
    {
        Console.WriteLine(((userInfo as Hashtable)["Camera"] as
Camera).CameraInstanceName.ToString() + " " +
                        "C-AF");
    }
}

```

If you want to stop receiving notification, call `removeNotificationObserver()`.

5. Obtain Camera Stream

- step1. Select a camera
- step2. Use `supportedFormats()` to get a list of camera video stream formats.
- step3. Use `selectedCamera.setFormat()`, to set a format chosen from the result of Step 2.
- step4. Use `addStreamObserverEX(selectedCamera, cb_streamObserverEX)`, to get a callback image data

```

Camera selectedCamera = cameraListComboBox.SelectedItem as Camera;
List<VideoResolutionFormat> cameraSupportVideoFormat = selectedCamera.supportedFormats();
selectedCamera.setFormat(cameraSupportVideoFormat[this.resolutionList.SelectedIndex]);

StreamProxy.StreamObserverEX cb_streamObserverEX = new
StreamProxy.StreamObserverEX(this.update_ucCameraModelImageEX);
StreamProxy.sharedProxy.addStreamObserverEX(selectedCamera, cb_streamObserverEX);

private void update_ucCameraModelImageEX(Camera camera, IntPtr pBitmap, int size, DirectShowLib.IMediaSample
mediaSample)
{
    //Process the Image
    updateImageContent(camera, pBitmap, size);
    return;
}

```

6. Camera Focus Command

```

Camera selectedCamera = cameraListComboBox.SelectedItem as Camera;
selectedCamera.setFocus(false, 0); //set autofocus mode
selectedCamera.startFocus();

```

7. Other camera function commands

```

selectedCamera.setFocus(false, 0);
selectedCamera.setAutoExposureLock(true);
selectedCamera.setExposure(short value);
selectedCamera.setAutoWhiteBalance(bool);
selectedCamera.setWhiteBalance(short value);
selectedCamera.setFrequency(PowerlineFrequency value)

selectedCamera.setBrightness((short)adjValue);
selectedCamera.setContrast((short)adjValue);
selectedCamera.setGamma((short)adjValue);
selectedCamera.setHue((short)adjValue);
selectedCamera.setSaturation((short)adjValue);
selectedCamera.setSharpness((short)adjValue)

selectedCamera.getFocusValue(out getValue, Camera.PropertyValueType);
selectedCamera.getAutoExposureLock(out boolValue);
selectedCamera.getExposure(out short value, PropertyValueType type)
selectedCamera.getAutoWhiteBalance(out bool value, PropertyValueType type)

```

```
selectedCamera.getWhiteBalance(out short value, PropertyValue type)
selectedCamera.getFrequency(out PowerlineFrequency value, PropertyValue type)

selectedCamera.getBrightness(out getValue, Camera.PropertyValueType);
selectedCamera.getContrast(out getValue, Camera.PropertyValueType);
selectedCamera.getGamma(out getValue, Camera.PropertyValueType);
selectedCamera.getHue(out getValue, Camera.PropertyValueType);
selectedCamera.getSaturation(out getValue, Camera.PropertyValueType);
selectedCamera.getSharpness(out getValue, Camera.PropertyValueType);
```