

Basler Pylon GigE provider

Version 1.2.0

User's guide

June 30, 2020

【 Remarks 】

【 Revision history 】

Version	Date	Content
1.0.0	2013-07-17	First edition.
1.0.1	2014-06-19	Added commands. PutUserSetSelector, GetUserSetSelector, PutExposureTimeRaw, GetExposureTimeRaw
1.1.0	2018-01-22	Supported windows10
	2018-09-20	Pylon5 version 5.0.12.
	2018-10-12	Added an error code that occurs when the camera's trigger mode is off.
	2019-10-29	Added Pylon5 setup procedure.
1.2.0	2020-06-01	Compatible with Pylon SDK Version 6

【 Hardware 】

Model	Version	Notes
ace640-100gm		
aceA1300-30gg		

Contents

1. Introduction.....	5
2. Outline of provider	6
2.1. Outline.....	6
2.2. Method and Property.....	7
2.2.1. CaoWorkspace::AddController method	7
2.2.2. CaoController::AddVariable method	8
2.2.3. CaoController::Execute method	8
2.2.4. CaoVariable::get_Value method	8
2.2.5. CaoVariable::put_Value method	8
2.3. Variable list.....	9
2.3.1. Controller class.....	9
2.4. Error code	9
3. Command reference.....	10
3.1. Controller class	10
3.1.1. CaoController::Execute("Start") command	11
3.1.2. CaoController::Execute("Stop") command.....	12
3.1.3. CaoController::Execute("GetImage") command	12
3.1.4. CaoController::Execute("GetLastErrorDescription") command.....	12
3.1.5. CaoController::Execute("ExecSoftTrigger") command	13
3.1.6. CaoController::Execute("GetSensorWidth") command	13
3.1.7. CaoController::Execute("GetSensorHeight") command.....	13
3.1.8. CaoController::Execute("GetWidthMax") command	14
3.1.9. CaoController::Execute("GetHeightMax") command	14
3.1.10. CaoController::Execute("GetDeviceVendorName") command	14
3.1.11. CaoController::Execute("GetDeviceModelName") command	15
3.1.12. CaoController::Execute("GetDeviceManufacturerInfo") command	15
3.1.13. CaoController::Execute("GetDeviceVersion") command	15
3.1.14. CaoController::Execute("GetDeviceFirmwareVersion") command.....	16
3.1.15. CaoController::Execute("GetDeviceID") command	16
3.1.16. CaoController::Execute("PutDeviceUserID") command.....	16
3.1.17. CaoController::Execute("GetDeviceUserID") command	17
3.1.18. CaoController::Execute("DeviceReset") command.....	17
3.1.19. CaoController::Execute("PutDeviceScanType") command.....	17
3.1.20. CaoController::Execute("GetDeviceScanType") command	18
3.1.21. CaoController::Execute("PutWidth") command	18
3.1.22. CaoController::Execute("GetWidth") command	18

3.1.23. CaoController::Execute("PutHeight") command	19
3.1.24. CaoController::Execute("GetHeight") command	19
3.1.25. CaoController::Execute("PutOffsetX") command	19
3.1.26. CaoController::Execute("GetOffsetX") command	20
3.1.27. CaoController::Execute("PutOffsetY") command	20
3.1.28. CaoController::Execute("GetOffsetY") command	20
3.1.29. CaoController::Execute("PutCenterX") command.....	21
3.1.30. CaoController::Execute("GetCenterX") command	21
3.1.31. CaoController::Execute("PutCenterY") command.....	21
3.1.32. CaoController::Execute("GetCenterY") command	22
3.1.33. CaoController::Execute("PutLineSelector") command.....	22
3.1.34. CaoController::Execute("GetLineSelector") command	23
3.1.35. CaoController::Execute("PutLineInverter") command.....	23
3.1.36. CaoController::Execute("GetLineInverter") command	23
3.1.37. CaoController::Execute("PutLineDebouncerTimeAbs") command	24
3.1.38. CaoController::Execute("GetLineDebouncerTimeAbs") command	24
3.1.39. CaoController::Execute("GetLineStatus") command	24
3.1.40. CaoController::Execute("GetLineStatusAll") command	25
3.1.41. CaoController::Execute("PutLineMode") command	25
3.1.42. CaoController::Execute("GetLineMode") command.....	25
3.1.43. CaoController::Execute("PutLineTermination") command	26
3.1.44. CaoController::Execute("GetLineTermination") command	26
3.1.45. CaoController::Execute("PutLineSource") command.....	26
3.1.46. CaoController::Execute("GetLineSource") command	27
3.1.47. CaoController::Execute("PutUserOutputSelector") command	27
3.1.48. CaoController::Execute("GetUserOutputSelector") command.....	28
3.1.49. CaoController::Execute("PutUserOutputValue") command	28
3.1.50. CaoController::Execute("GetUserOutputValue") command	28
3.1.51. CaoController::Execute("GetUserOutputValueAll") command	29
3.1.52. CaoController::Execute("PutUserSetSelector") Command.....	29
3.1.53. CaoController::Execute("GetUserSetSelector") Command	30
3.1.54. CaoController::Execute("PutExposureTimeRaw") Command.....	31
3.1.55. CaoController::Execute("GetExposureTimeRaw") Command	31
Appendix A. Pylon 5 (SDK Version 5.0.12) Setup Procedure.....	32
Appendix B. Pylon 6 (SDK Version 6.1.0) Setup Procedure	37
Appendix C. Table for Pylon API.....	40

1. Introduction

This book is a user's guide of the PylonGigE provider that is one of the CAO providers.

The PylonGigE provider communicates with the GigE Vision camera (henceforth GigE camera) with the camera control software Pylon manufactured by Basler.

To use the PylonGigE provider, one of the following Basler camera control software is required.

- Pylon SDK Version 4.0.0
- Pylon SDK Version 5.0.12
- Pylon SDK Version 6.1.0

Please note that Versoin4 series cannot be installed on Windows 10 OS.

The next chapter describes the outline of the PylonGigE provider, and Chapter 3 lists command references.

2. Outline of provider

2.1. Outline

The PylonGigE provider is CAO provider that absorbs the dependence on Pylon API and then provides functions specified by the CAO provider interface specification. The communication with the GigE camera uses the CBaslerGigEInstantCamera class of Pylon.

When starting, the PylonGigE provider loads the default start-up setting of the GigE camera. For the way of default start-up, refer to "C++ Programmer's Guide and Reference Documentation" of the camera control software Pylon manufactured by Basler.

The file format is DLL (Dynamic Link Library), and it is dynamically uploaded from the CAO engine when it is used. To use PylonGigE provider, you need to install ORiN2SDK, or, complete registration manually with the procedure shown below.

Table 2-1 PylonGigE provider

File name	CaoProvPylonGigE.dll
ProgID	CaoProv.Basler.PylonGigE
Registry registration	regsvr32 CaoProvPylonGigE.dll
Remove registry registration	regsvr32 /u CaoProvPylonGigE.dll

Moreover, the PylonGigE provider needs the registry registration of the provider by the installation situation of PC used.

Table 2-2 Additional provider registration

Installed Pylon SDK Version 4.0.0	regsvr32 CaoProvPylon4GigE.dll
Installed Pylon SDK Version 5.0.12	regsvr32 CaoProvPylon5GigE.dll
Installed Pylon SDK Version 6.1.0	regsvr32 CaoProvPylon6GigE.dll

priority is used, the option with low priority is disregarded. The priority of the option is as follows.

Conn option > Serial option > UserID option

※Note

Be sure to set the camera's trigger mode to ON. AddController will fail if the trigger mode is OFF.

2.2.2. CaoController::AddVariable method

Create a variable object.

Please refer to 2.3.1 for the variable mounted in the PylonGigE provider.

Syntax AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])
 <bstrVariableName > : [in] Variable name
 <bstrOption> : [in] Option string

Example

```
Dim aaa As Object
Dim bbb As Double

Set aaa = caoCtrl.AddVariable("@IMAGE")
bbb = aaa.Value
```

2.2.3. CaoController::Execute method

Execute a command. For arguments of Execute method, specify a command with BSTR, specify a parameter with a VARIANT array. Please refer to 3.1 for details of each command.

Format [<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])
 < vntRet > : [out] Return value of command
 < bstrCmd > : [in] Command
 < vntParam > : [in] Parameter

2.2.4. CaoVariable::get_Value method

Acquire a variable value.

Please refer to 2.3 for details of the acquired value.

2.2.5. CaoVariable::put_Value method

Set a variable value.

Please refer to 2.3 for details of the set value.

2.3. Variable list

2.3.1. Controller class

Table 2-4 Controller class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@IMAGE	VT_I4	Acquisition of image.	√	-

2.4. Error code

In the PylonGigE provider, specific error codes shown below are designated. Please refer to the chapter of the error code of "[ORiN2 programming guide](#)" for the ORiN2 common errors.

Table 2-5 Specific error code list

Error name	Error number	Explanation
E_CAOP_PYLON_ERR	0x80100001	An error occurs on Pylon API. Please acquire details in "GetLastErrorDescription" of CaoController::Execute().
E_CAOP_UNINIT	0x80100002	Not initialized status.
E_CAOP_NOIMAGE	0x80100003	No image exists.
E_CAOP_DEVICE_NOTFOUND	0x80100004	No camera device is found.
E_CAOP_DEVICE_INVALID	0x80100005	The specification of the camera device is incorrect.
E_CAOP_TRIGGERMODE_OFF	0x80100006	Camera's trigger mode is not "ON".
E_CAOP_VERSION_INVALID	0x80100006	The version specification is incorrect.

3. Command reference

This chapter explains each command of the `CaoController::Execute` method.

Each command of the `CaoController::Execute` method executes `CBaslerGigEInstantCamera` class member of Pylon API. Please refer to Appendix A for the correspondence of each command and PylonAPI.

For detailed operation, refer to "C++ Programmer's Guide and Reference Documentation" of the camera control software Pylon manufactured by Basler.

3.1. Controller class

Table 3-1 CaoController::Execute command list

Command	Function	
Start	Start image shooting	P. 11
Stop	Stop image shooting	P. 12
GetImage	Acquire the latest image	P. 12
GetLastErrorDescription	Acquire detail of Pylon API errors	P. 12
ExecSoftTrigger	Execute software trigger	P. 13
GetSensorWidth	Acquire width of camera sensor	P. 13
GetSensorHeight	Acquire height of camera sensor	P. 13
GetWidthMax	Acquire the maximum width	P. 14
GetHeightMax	Acquire the maximum height	P. 14
GetDeviceVendorName	Acquire vender name	P. 14
GetDeviceModelName	Acquire model name	P. 15
GetDeviceManufacturerInfo	Acquire additional information from manufacturer	P. 15
GetDeviceVersion	Acquire version	P. 15
GetDeviceFirmwareVersion	Acquire firmware version	P. 16
GetDeviceID	Acquire Device ID	P. 16
PutDeviceUserID	Set user ID	P. 16
GetDeviceUserID	Acquire user ID	P. 17
DeviceReset	Reset a device	P. 17
PutDeviceScanType	Set scan type of device sensor	P. 17
GetDeviceScanType	Acquire scan type of device sensor	P. 18
PutWidth	Set width of AOI	P. 18
GetWidth	Acquire width of AOI	P. 18
PutHeight	Set height of AOI	P. 19
GetHeight	Acquire height of AOI	P. 19
PutOffsetX	Set X-offset of AOI	P. 19
GetOffsetX	Acquire X-offset of AOI	P. 20
PutOffsetY	Set Y-offset of AOI	P. 20
GetOffsetY	Acquire Y-offset of AOI	P. 20

PutCenterX	Set horizontal centering flag	P. 21
GetCenterX	Acquire horizontal centering flag	P. 21
PutCenterY	Set vertical centering flag	P. 21
GetCenterY	Acquire vertical centering flag	P. 22
PutLineSelector	Switch the currently selected I/O	P. 22
GetLineSelector	Acquire the currently selected I/O	P. 23
PutLineInverter	Switch ON/OFF the currently selected I/O's signal inverter function	P. 23
GetLineInverter	Acquire the status of the currently selected I/O's signal inverter function	P. 23
PutLineDebouncerTimeAbs	Set absolute value for the debouncer time of currently selected I/O	P. 24
GetLineDebouncerTimeAbs	Acquire the absolute value of the debouncer time of currently selected I/O	P. 24
GetLineStatus	Acquire logical status of the currently selected I/O	P. 24
GetLineStatusAll	Acquire logical state of all I/O	P. 25
PutLineMode	Set mode of the currently selected I/O	P. 25
GetLineMode	Acquire mode of the currently selected I/O	P. 25
PutLineTermination	Switch ON/OFF the terminator function of currently selected Input	P. 26
GetLineTermination	Acquire the terminator function status of currently selected Input	P. 26
PutLineSource	Set the source signal of currently selected Output	P. 26
GetLineSource	Acquire the source signal of currently selected Output	P. 27
PutUserOutputSelector	Set the currently selected user output	P. 27
GetUserOutputSelector	Acquire the currently selected user output	P. 28
PutUserOutputValue	Set the value of currently selected user output	P. 28
GetUserOutputValue	Output the value of currently selected user output	P. 28
GetUserOutputValueAll	Acquire the value of all user outputs	P. 29
PutUserSetSelector	Set the number of currently selected userset.	P. 29
GetUserSetSelector	Acquire the number of currently selected userset.	P. 30
PutExposureTimeRaw	Set the value of currently exposure parameter.	P. 31
GetExposureTimeRaw	Acquire the value of currently exposure parameter.	P. 31

3.1.1. CaoController::Execute("Start") command

Start image shooting with the GigE camera.

Syntax Start()

Return value : None

example

```
caoCtrl.Start ' Start image shooting
```

3.1.2. CaoController::Execute("Stop") command

Stop image shooting with the GigE camera.

Clear the shot image.

Syntax Stop()

Return value : None

example

```
caoCtrl.Stop 'Stop image shooting
```

3.1.3. CaoController::Execute("GetImage") command

Acquire image shot by the GigE camera.

If an image has not acquired yet, the command waits for a certain period of time specified by Time option of AddContorller() for receiving a image.

When this command succeeds, an image obtained will be cleared. a

Syntax GetImage()

Return value : [out] Image (VT_UI1 | VT_ARRAY)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetImage 'Acquire image
```

3.1.4. CaoController::Execute("GetLastErrorDescription") command

Acquire error details of Pylon API that has occurred in the last.

When the error of Pylon API occurs, E_CAOP_PYLON_ERR (0x80100001) is generated. With executing this command immediately after the error occurrence, you can check the error details.

Syntax GetLastErrorDescription ()

Return value : [out] Detailed Error(VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLastErrorDescription 'Acquire error details
```

3.1.5. CaoController::Execute("ExecSoftTrigger") command

Execute software trigger for the GigE camera.

Syntax ExecSoftTrigger()

Return value : None

Example

```
caoCtrl.ExecSoftTrigger 'Execute Software trigger
```

3.1.6. CaoController::Execute("GetSensorWidth") command

Acquire the width of the sensor of the GigE camera.

Syntax GetSensorWidth()

Return value : [out] Width (VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetSensorWidth 'Acquire width of sensor
```

3.1.7. CaoController::Execute("GetSensorHeight") command

Acquire the height of the sensor of the GigE camera.

Syntax GetSensorHeight()

Return value : [out] Height (VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetSensorHeight 'Acquire sensor amount
```

3.1.8. CaoController::Execute("GetWidthMax") command

Acquire the maximum value of the width of the GigE camera.

Syntax GetWidthMax()

Return value : [out] Width (VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetWidthMax 'Acquire the maximum value of width
```

3.1.9. CaoController::Execute("GetHeightMax") command

Acquire the height of the GigE camera sensor.

Syntax GetHeightMax()

Return value : [out] Height (VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetHeightMax 'Acquire the maximum value of height
```

3.1.10. CaoController::Execute("GetDeviceVendorName") command

Acquire vender of the GigE camera.

Syntax GetDeviceVendorName()

Return value : [out] Vender name (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceVendorName 'Acquire vender name
```

3.1.11. CaoController::Execute("GetDeviceModelName") command

Acquire the model name of the GigE camera.

Syntax GetDeviceModelName()

Return value : [out] Model name (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceModelName 'Acquire model name
```

3.1.12. CaoController::Execute("GetDeviceManufacturerInfo") command

Acquire additional information from the manufacturer of the GigE camera.

Syntax GetDeviceManufacturerInfo()

Return value : [out] Additional information (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceManufacturerInfo 'Acquire additional information
```

3.1.13. CaoController::Execute("GetDeviceVersion") command

Acquire the device version of the GigE camera.

Syntax GetDeviceVersion()

Return value : [out] Device version (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceVersion 'Acquire device version
```

3.1.14. CaoController::Execute("GetDeviceFirmwareVersion") command

Acquire the firmware version of the GigE camera.

Syntax GetDeviceFirmwareVersion()

Return value : [out] Firmware version (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceFirmwareVersion 'Acquire firmware version
```

3.1.15. CaoController::Execute("GetDeviceID") command

Acquire Device ID of the GigE camera.

Syntax GetDeviceID()

Return value : [out] Device ID (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceID 'Acquire Device ID
```

3.1.16. CaoController::Execute("PutDeviceUserID") command

Set user ID of the GigE camera.

Syntax PutDeviceUserID(<bstrUserID:VT_BSTR>)

< bstrUserID > : [in] User ID(VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutDeviceUserID("Sample") 'Set user ID
```

3.1.17. CaoController::Execute("GetDeviceUserID") command

Acquire user ID of the GigE camera.

Syntax GetDeviceUserID()

Return value : [out] User ID (VT_BSTR)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceUserID 'Acquire user ID
```

3.1.18. CaoController::Execute("DeviceReset") command

Reset the GigE camera.

Executing this command disconnects the GigE camera. To connect the GigE camera again, discard the current CaoController object, and then retry AddController().

Syntax DeviceReset()

Return value : None

Example

```
caoCtrl.DeviceReset ' Reset execution
```

3.1.19. CaoController::Execute("PutDeviceScanType") command

Set the scan type of the device sensor of the GigE camera.

Syntax PutDeviceScanType(<IScanType:VT_BSTR>)

<IScanType> : [in] Scanning type(VT_I4)

Use the value of Basler_GigECamera::DeviceScanTypeEnums of PylonAPI

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutDeviceUserID(0) 'Set scan type
```

3.1.20. CaoController::Execute("GetDeviceScanType") command

Acquire the scan type of the device sensor of the GigE camera.

Syntax GetDeviceScanType()

Return value : [out] Scanning type(VT_I4)

Use the value of Basler_GigECamera::DeviceScanTypeEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceScanType 'Acquire scan type
```

3.1.21. CaoController::Execute("PutWidth") command

Set the width of AOI.

Syntax PutWidth(<IWidth:VT_BSTR>)

<IScanType> : [in] Width(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutWidth(320) 'Setting of width
```

3.1.22. CaoController::Execute("GetWidth") command

Acquire the width of AOI.

Syntax GetWidth()

Return value : [out] Width(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetWidth 'Acquire width
```

3.1.23. CaoController::Execute("PutHeight") command

Set the height of AOI.

Syntax PutHeight(<IHeight:VT_BSTR>)

<IScanType> : [in] Height(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutHeight(240) 'Set the height
```

3.1.24. CaoController::Execute("GetHeight") command

Acquire the height of AOI.

Syntax GetHeight()

Return value : [out] Height(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetHeight 'Acquire the height
```

3.1.25. CaoController::Execute("PutOffsetX") command

Set X-offset of AOI.

Syntax PutOffsetX(<IOffsetX:VT_BSTR>)

<IOffsetX> : [in] X offset(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutOffsetX(100) 'Set the X-offset
```

3.1.26. CaoController::Execute("GetOffsetX") command

Acquire X-offset of AOI.

Syntax GetOffsetX()

Return value : [out] X offset(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetOffsetX 'Acquire the X-offset
```

3.1.27. CaoController::Execute("PutOffsetY") command

Set Y-offset of AOI.

Syntax PutOffsetY(<IOffsetY:VT_BSTR>)

<IOffsetY> : [in] Y offset(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutOffsetY(100) 'Set the Y-offset
```

3.1.28. CaoController::Execute("GetOffsetY") command

Acquire Y-offset of AOI.

Syntax GetOffsetY()

Return value : [out] Y offset(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetOffsetY 'Acquire the Y-offset
```

3.1.29. CaoController::Execute("PutCenterX") command

Set the horizontal centering flag of AOI.

Syntax PutCenterX(<lCenterX:VT_BSTR>)

<lCenterX> : [in] Centering flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutCenterX(100) 'Set the centering flag
```

3.1.30. CaoController::Execute("GetCenterX") command

Acquire the horizontal centering flag of AOI.

Syntax GetCenterX()

Return value : [out] Centering flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetCenterX 'Acquire the centering flag
```

3.1.31. CaoController::Execute("PutCenterY") command

Set the vertical centering flag of AOI.

Syntax PutCenterY(<ICenterY:VT_BSTR>)

<ICenterY> : [in] Centering flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutCenterY(100) 'Setting of centering flag
```

3.1.32. CaoController::Execute("GetCenterY") command

Acquire the vertical centering flag of AOI.

Syntax GetCenterY()

Return value : [out] Centering flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetCenterY 'Acquire the centering flag
```

3.1.33. CaoController::Execute("PutLineSelector") command

Switch the currently selected I/O.

Syntax PutLineSelector(<IIO:VT_I4>)

<IIO> : [in] Selection I/O(VT_I4)

Use the value of Basler_GigECamera::LineSelectorEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineSelector(0) 'Switch the currently selected I/O
```

3.1.34. CaoController::Execute("GetLineSelector") command

Acquire currently selected I/O.

Syntax GetLineSelector()

Return value : [out] Selection I/O(VT_I4)

Use the value of Basler_GigECamera::LineSelectorEnums of PylonAPI .

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineSelector 'Acquire the currently selected I/O
```

3.1.35. CaoController::Execute("PutLineInverter") command

Switch ON/OFF of the signal inverter function of currently selected I/O.

Syntax PutLineInverter(<IEnable:VT_BOOL>)

<IEnable> : [in] Signal inverter function flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineInverter(false) 'Signal inverter function setting
```

3.1.36. CaoController::Execute("GetLineInverter") command

Acquire the ON/OFF status of the currently selected I/O's signal inverter function.

Syntax GetLineInverter()

Return value : [out] Signal inverter function flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineInverter 'Acquire signal inverter function state
```

3.1.37. CaoController::Execute("PutLineDebouncerTimeAbs") command

Set the absolute value of the debouncer time of the currently selected I/O (Unit: microsecond)

Syntax PutLineDebouncerTimeAbs(<dTime:VT_BOOL>)

<dTime> : [in] debouncer time(VT_R8)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineDebouncerTimeAbs(10.00) 'Set the debouncer time
```

3.1.38. CaoController::Execute("GetLineDebouncerTimeAbs") command

Acquire the absolute value of the debouncer time of the currently selected I/O (Unit: microsecond)

Syntax GetLineDebouncerTimeAbs()

Return value : [out] debouncer time(VT_R8)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineDebouncerTimeAbs 'Acquire debouncer time
```

3.1.39. CaoController::Execute("GetLineStatus") command

Acquire the logical state of the currently selected I/O.

Syntax GetLineStatus()

Return value : [out] State(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineStatus 'State acquisition
```

3.1.40. CaoController::Execute("GetLineStatusAll") command

Acquire the logical state of all I/O.

Syntax GetLineStatusAll()

Return value : [out] State(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineStatusAll 'State acquisition
```

3.1.41. CaoController::Execute("PutLineMode") command

Set the mode of the currently selected I/O.

Syntax PutLineMode(<IMode:VT_I4>)

<IMode> : [in] Mode(VT_I4)

Use the value of Basler_GigECamera::LineModeEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineMode(0) 'Mode setting
```

3.1.42. CaoController::Execute("GetLineMode") command

Acquire the mode of the currently selected I/O.

Syntax GetLineMode()

Return value : [out] Mode(VT_I4)

Use the value of Basler_GigECamera::LineModeEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineMode 'Mode acquisition
```

3.1.43. CaoController::Execute("PutLineTermination") command

Switch ON/OFF the terminator function of the currently selected Input.

Syntax PutLineTermination(<IEnable:VT_BOOL>)

<IEnable> : [in] Terminator function flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineTermination(false) 'Terminator function setting
```

3.1.44. CaoController::Execute("GetLineTermination") command

Acquire the ON/OFF status of the terminator function of the currently selected Input.

Syntax GetLineTermination()

Return value : [out] Terminator function flag(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineTermination 'Acquire terminator function state
```

3.1.45. CaoController::Execute("PutLineSource") command

Set the source signal of the currently selected Output.

Syntax PutLineSource(<ISource:VT_I4>)

<ISource> : [in] Source signal(VT_I4)

Use value of Basler_GigECamera::LineSourceEnums of PylonAPI

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutLineSource(0) 'Mode setting
```

3.1.46. CaoController::Execute("GetLineSource") command

Acquire the source signal of the currently selected Output.

Syntax GetLineSource()

Return value : [out] Source signal(VT_I4)
 Use value of Basler_GigECamera::LineSourceEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetLineSource 'Mode acquisition
```

3.1.47. CaoController::Execute("PutUserOutputSelector") command

Set the currently selected user output

Syntax PutUserOutputSelector(<IUserOutput:VT_I4>)

<IUserOutput> : [in] Selection user output(VT_I4)
 Use the value of
 Basler_GigECamera::SyncUserOutputSelectorEnums of PylonAPI.

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutUserOutputSelector(0) 'Set User output
```

3.1.48. **GaoController::Execute("GetUserOutputSelector")** command

Acquire the currently selected user output.

Syntax GetUserOutputSelector()

Return value : [out] User output(VT_I4)

Use value of Basler_GigECamera::SyncUserOutputSelectorEnums
of PylonAPI

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetUserOutputSelector 'Acquire user output
```

3.1.49. **GaoController::Execute("PutUserOutputValue")** command

Set the value of the currently selected user output.

Syntax PutUserOutputValue(<bVal:VT_BOOL>)

<bVal> : [in] Value(VT_BOOL)

Example

```
Dim vRes As Variant
vRes = caoCtrl.PutUserOutputValue(false) 'Set user output value
```

3.1.50. **GaoController::Execute("GetUserOutputValue")** command

Acquire the value of the currently selected user output.

Syntax GetUserOutputValue()

Return value : [out] Value(VT_BOOL)

Example

```
Dim vRes As Variant
```

```
vRes = caoCtrl.GetUserOutputValue 'Acquire user output value
```

3.1.51. CaoController::Execute("GetUserOutputValueAll") command

Acquire values of all user outputs.

Syntax GetUserOutputValueAll()

Return value : [out] Value(VT_I4)

Example

```
Dim vRes As Variant
vRes = caoCtrl.GetUserOutputValueAll 'Acquire user output values
```

3.1.52. CaoController::Execute("PutUserSetSelector") Command

Set the number of currently selected userset.

Syntax PutUserSetSelector(<IParam>)

<IParam> : [in] user set number(VT_I4)

For Pylon SDK Version 4

0 : Selects the default configuration set

1 : High gain factory set

2 : Factory set using auto functions

3 : Factory set enabling color adjustments

4 : Factory set enabling custom 0 settings

5 : Factory set enabling custom 1 settings

6 : Selects user set 1

7 : Selects user set 2

8 : Selects user set 3

For Pylon SDK Version 5, 6

0 : Selects the default configuration set

1 : High gain factory set

2 : Factory set using auto functions

3 : Factory set enabling color adjustments

4 : Factory set disabling color adjustments

- 5 : Factory set enabling custom 0 settings
- 6 : Factory set enabling custom 1 settings
- 7 : Selects user set 1
- 8 : Selects user set 2
- 9 : Selects user set 3
- 10: The Light Microscopy factory set can be loaded.(Version6 Only)

Example

```
caoCtrl.PutUserSetSelector 2
```

3.1.53. CaoController::Execute("GetUserSetSelector") Command

Acquire the number of currently selected userset.

Syntax GetUserSetSelector()

Return value : [out] User set number(VT_I4)

For Pylon SDK Version 4

- 0 : Selects the default configuration set
- 1 : High gain factory set
- 2 : Factory set using auto functions
- 3 : Factory set enabling color adjustments
- 4 : Factory set enabling custom 0 settings
- 5 : Factory set enabling custom 1 settings
- 6 : Selects user set 1
- 7 : Selects user set 2
- 8 : Selects user set 3

For Pylon SDK Version 5, 6

- 0 : Selects the default configuration set
- 1 : High gain factory set
- 2 : Factory set using auto functions
- 3 : Factory set enabling color adjustments
- 4 : Factory set disabling color adjustments
- 5 : Factory set enabling custom 0 settings
- 6 : Factory set enabling custom 1 settings
- 7 : Selects user set 1

- 8 : Selects user set 2
- 9 : Selects user set 3
- 10: The Light Microscopy factory set can be loaded.(Version6 Only)

Example

```
Dim IRet As Long
IRet = caoCtrl.GetUserSetSelector
```

3.1.54. CaoController::Execute("PutExposureTimeRaw") Command

Set the value of currently exposure parameter.

Syntax PutExposureTimeRaw(<IParam>)

<IParam> : [in] Value (VT_I4)

Example

```
caoCtrl.PutExposureTimeRaw 1000
```

3.1.55. CaoController::Execute("GetExposureTimeRaw") Command

Acquire the value of currently exposure parameter.

Syntax GetExposureTimeRaw()

Return value : [out] Value (VT_I4)

Example

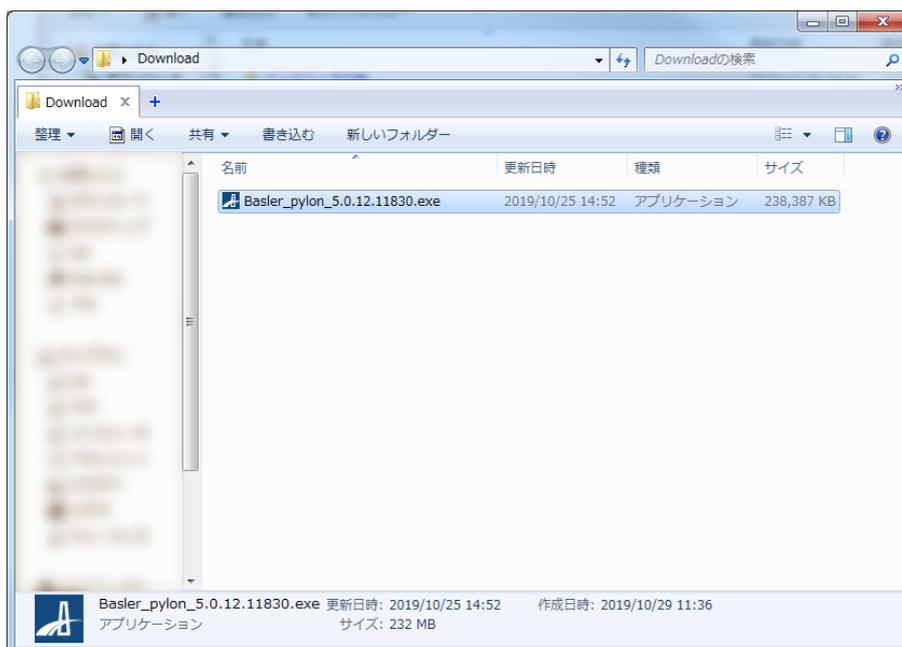
```
Dim IRet as Long
IRet = caoCtrl.GetExposureTimeRaw
```

Appendix A. Pylon 5 (SDK Version 5.0.12) Setup Procedure

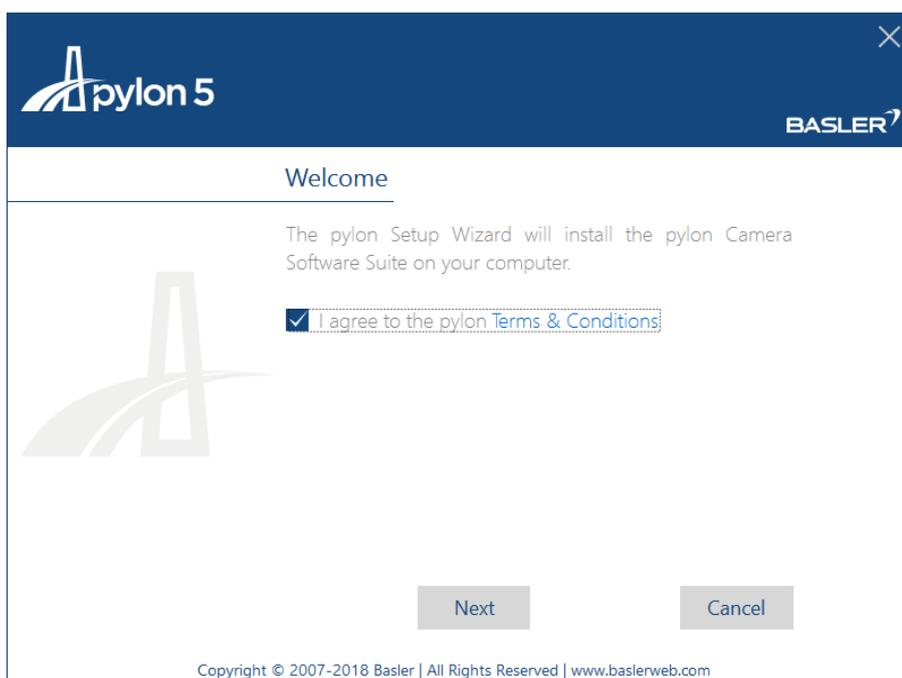
Describe the setup procedure of 5.0.12 which is the latest version of Pylon SDK Version 5.0.x.

1. SDK installation

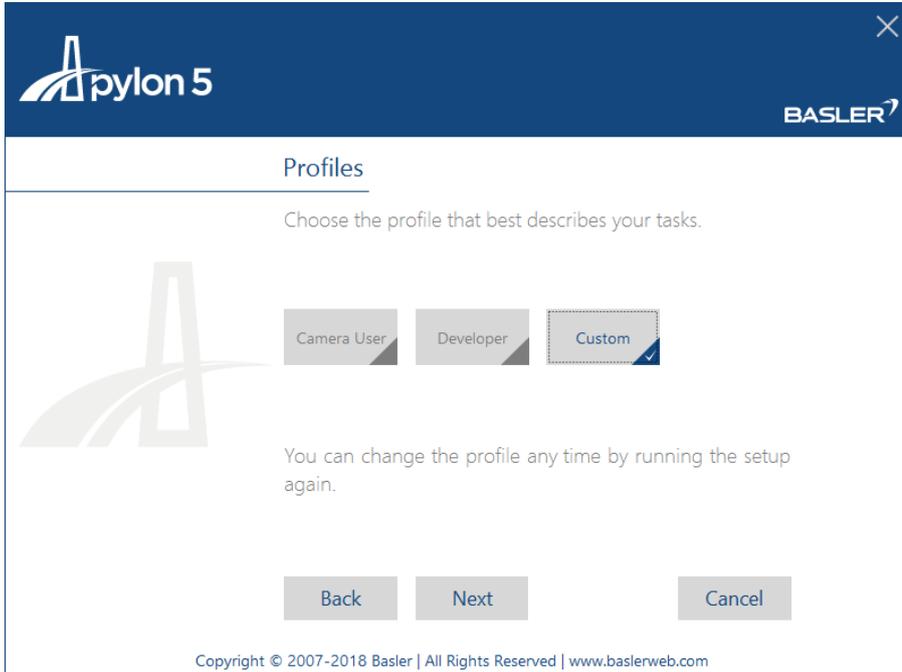
- Download the software suite Pylon 5.0.12 for Windows from Basler.
- Run the downloaded “Basler_pylon_5.0.12.11830.exe”.



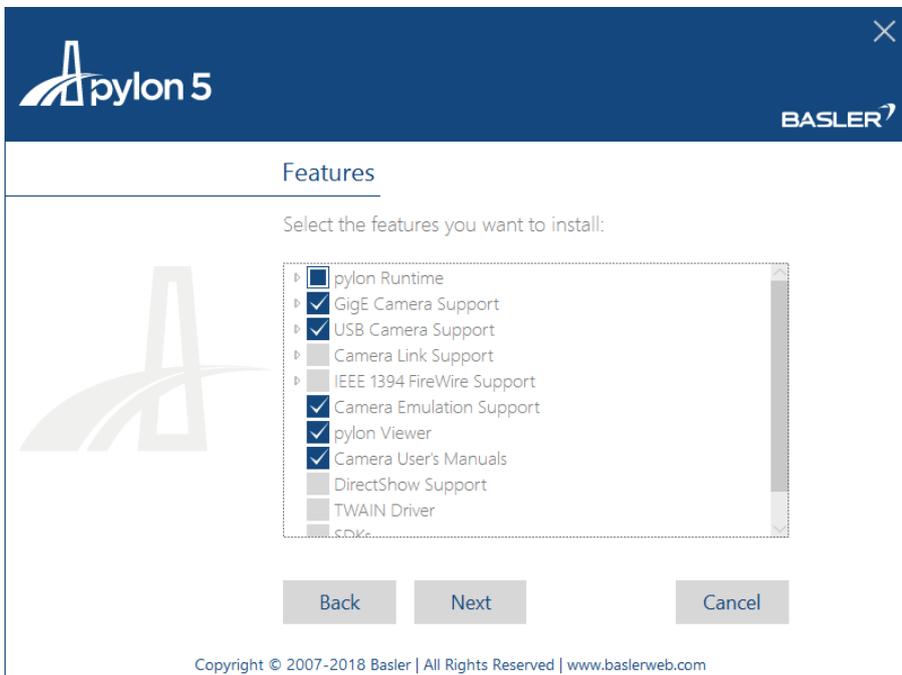
- Please check the agreement and proceed to the next.



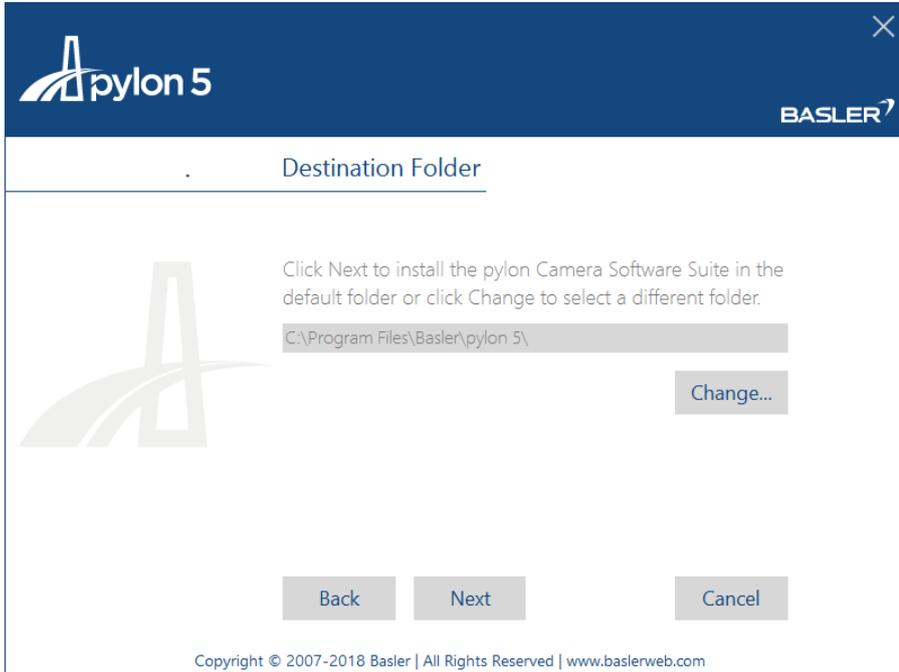
- Select “Custom” and continue.



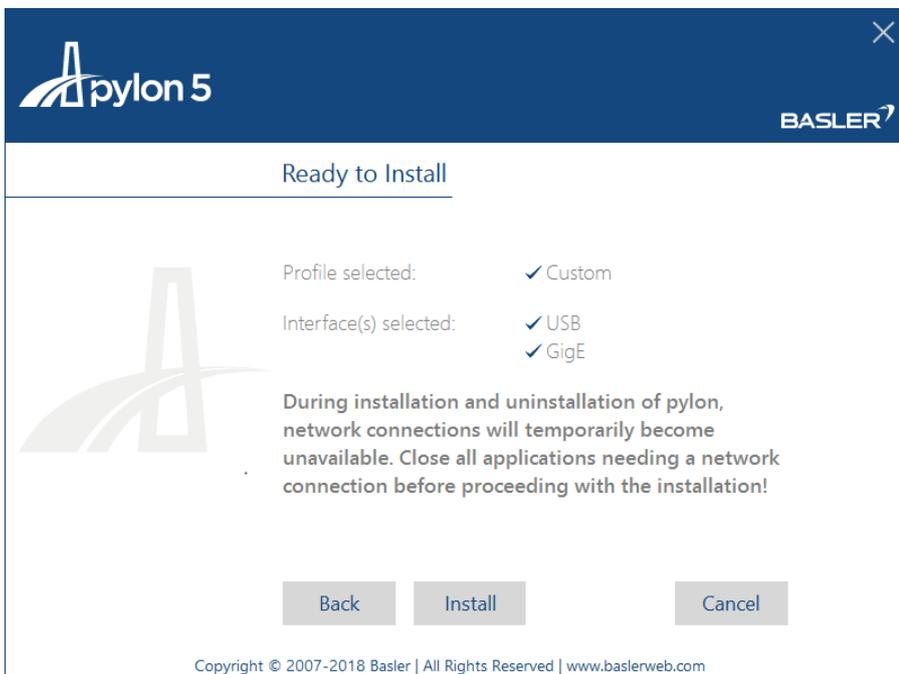
- Uncheck “DirectShow Support”, “TWAIN Driver”, and “Firmware Updater” from the functions checked by default, and proceed to the next.



- If necessary, change the installation directory and proceed to the next.

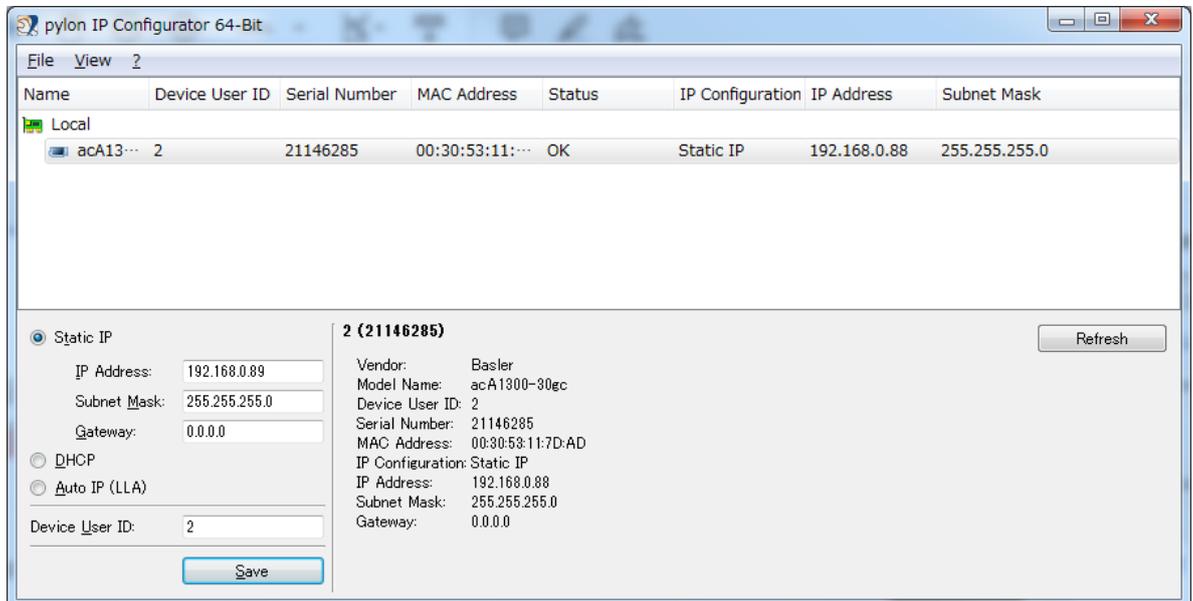


- Run the installation.



2. GigE camera IP address and ID settings

- Start the installed pylon IP Configurator, enter the “IP Address”, “Subnet Mask”, “Gateway”, “Device User ID” of the camera you want to set, and press the “Save” button.



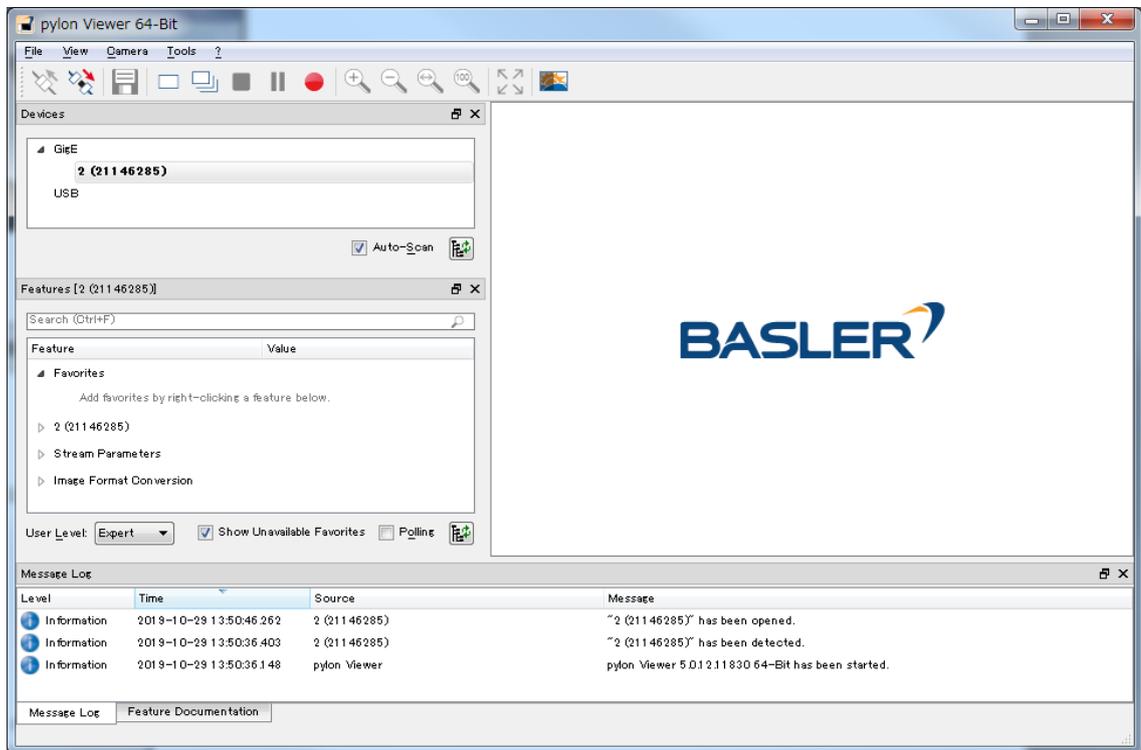
The range of Device User ID is 0-4.

User ID 0-4 corresponds to ID201-205 of the controller image storage area.

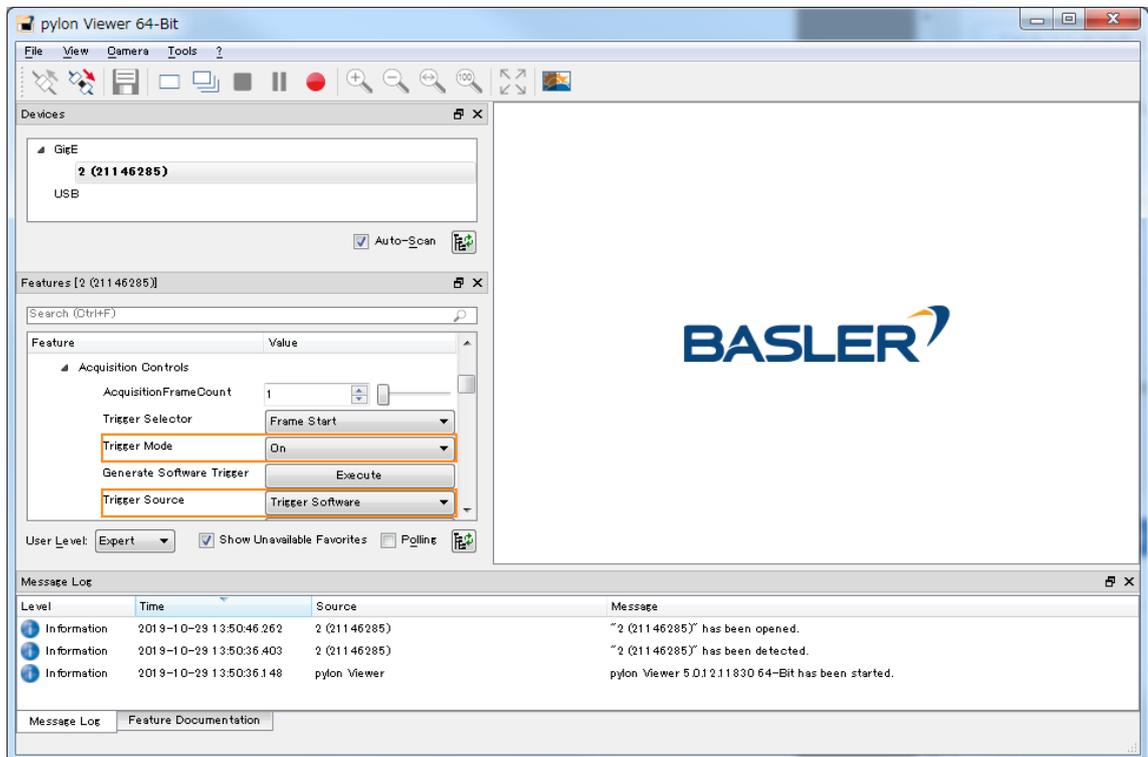
When using multiple cameras, do not duplicate IDs

3. Camera operation settings

- Start Pylon Viewer, select a camera, and set “User Level” to “Expert” or higher.

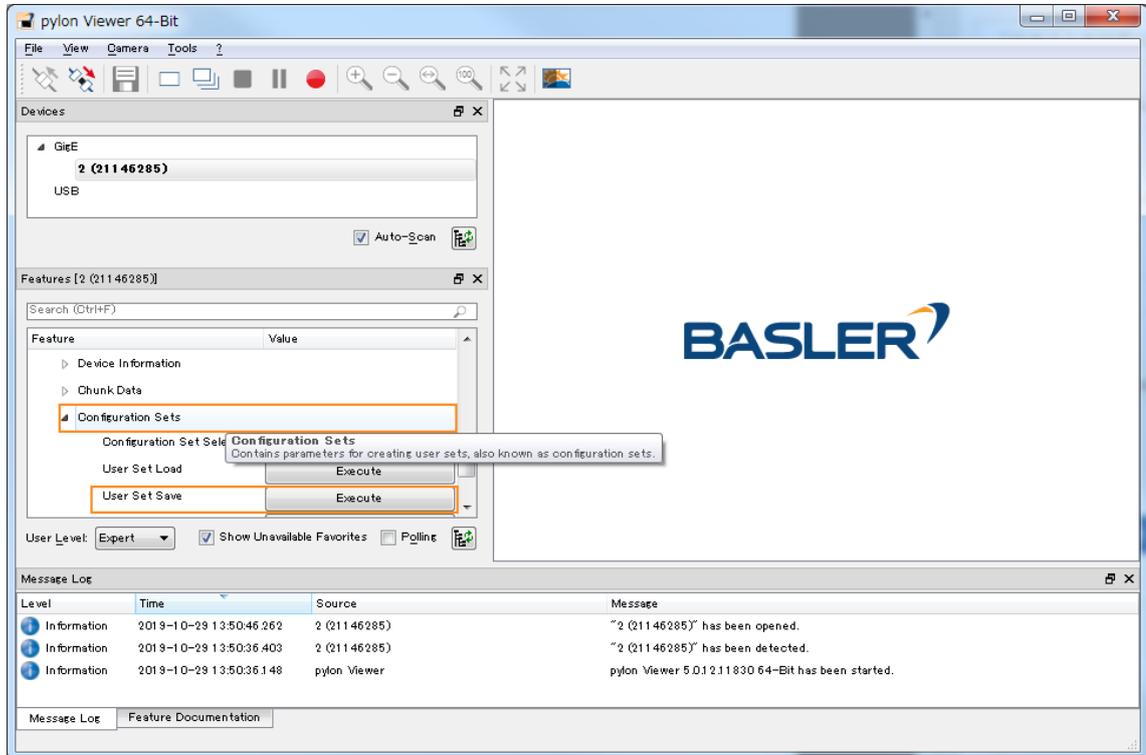


- Open “Feature” → “Acquisition Controls”.
Set “Trigger Mode” to “On” and select “Trigger Source” according to the mode to be used.



- Set other usage conditions (shutter, speed, gain, etc.).

- Use “Configuration Set Selector” to select the set to save the settings. Press “Execute” of “User Set Save” to save the set conditions. If this operation is not performed, the camera settings will not be saved.



When the camera is turned on, it starts up with the setting specified in “Default Startup Set” (“User Set2” in the figure above).

Appendix B. Pylon 6 (SDK Version 6.1.0) Setup Procedure

Describe the setup procedure of 6.1.0 which is the latest version of Pylon SDK Version 6.1.x.

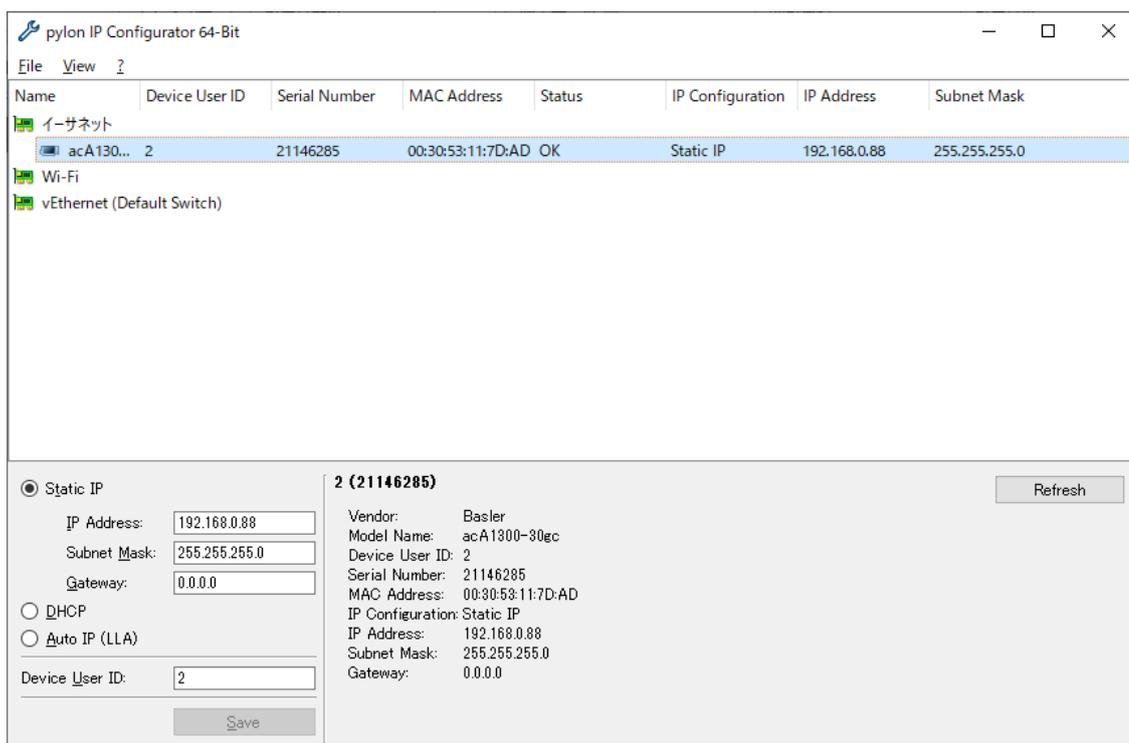
1. SDK installation

- Download the software suite Pylon 6.1.0 for Windows from Basler.
- Run the downloaded “Basler_pylon_6.1.0.19674.exe”.

See Appendix A for download and installation instructions.

2. GigE camera IP address and ID settings

- Start the installed pylon IP Configurator, enter the “IP Address”, “Subnet Mask”, “Gateway”, “Device User ID” of the camera you want to set, and press the “Save” button.



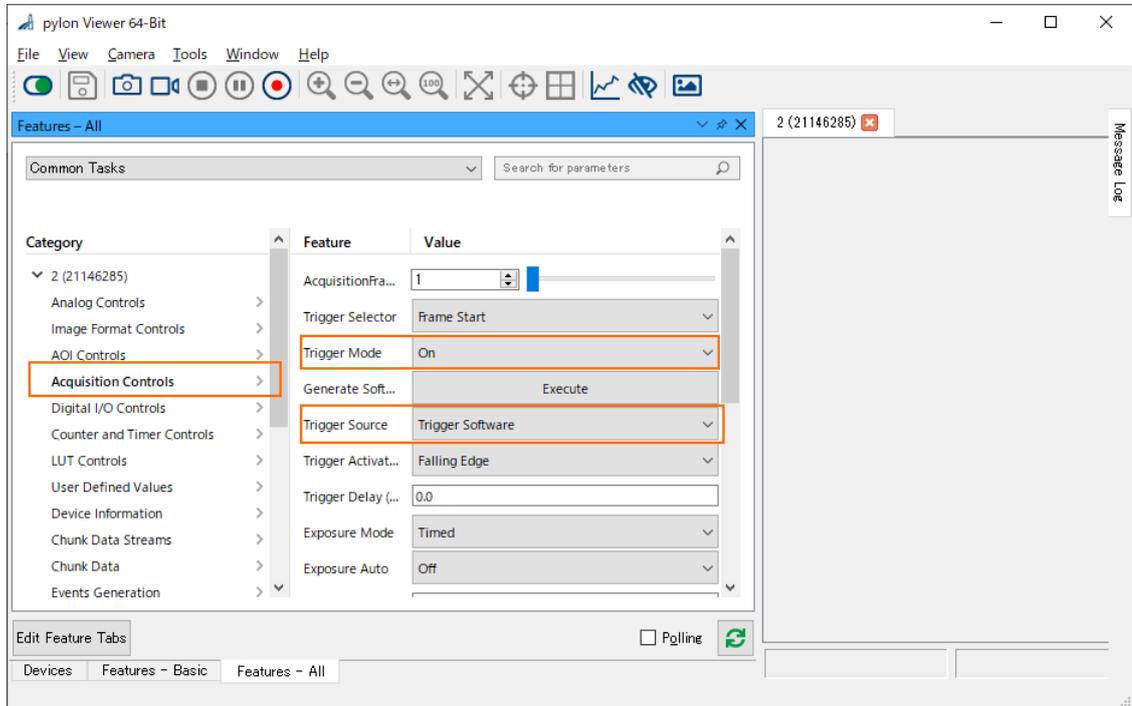
The range of Device User ID is 0-4.

User ID 0-4 corresponds to ID201-205 of the controller image storage area.

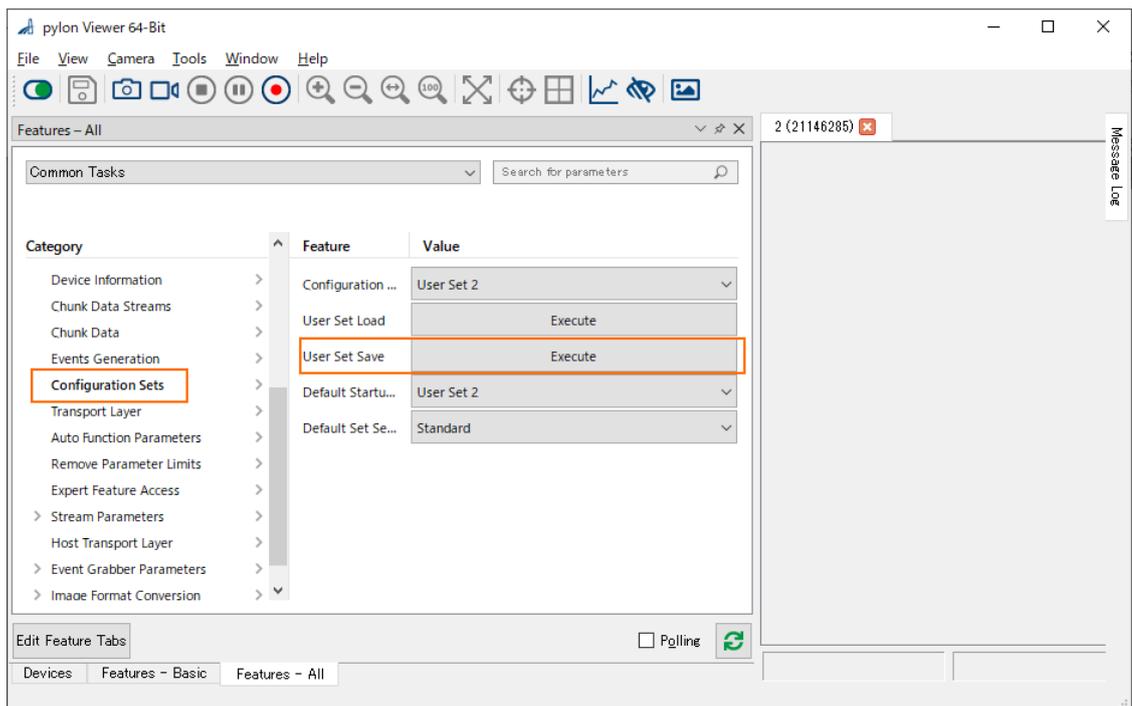
When using multiple cameras, do not duplicate IDs

3. Camera operation settings

- Open “Feature All” → “Acquisition Controls”. Set “Trigger Mode” to “On” and select “Trigger Source” according to the mode to be used.



- Set other usage conditions (shutter, speed, gain, etc.).
- Use “Configuration Set Selector” to select the set to save the settings. Press “Execute” of “User Set Save” to save the set conditions. If this operation is not performed, the camera settings will not be saved.



When the camera is turned on, it starts up with the setting specified in “Default Startup Set” (“User Set2” in the figure above).

Appendix C. Table for Pylon API

The PylonGigE provider communicates with the GigE camera by using the `CBaslerGigEInstantCamera` class of Pylon API.

This appendix shows the correspondence with each command of `CaoController::Execute()` and the `CBaslerGigEInstantCamera` class member.

Table 3–2 Table for `CaoController::Execute` method `CBaslerGigEInstantCamera` class

Execute command	<code>CBaslerGigEInstantCamera</code> class member
Start	<code>StartGrabbing</code>
Stop	<code>StopGrabbing</code>
GetImage	<code>RetrieveResult</code> (When image shooting). <code>GrabOne</code> (When image shooting stops).
<code>GetLastErrorDescription</code>	-
<code>ExecSoftTrigger</code>	<code>ExecuteSoftwareTrigger</code>
<code>GetSensorWidth</code>	<code>SensorWidth</code>
<code>GetSensorHeight</code>	<code>SensorHeight</code>
<code>GetWidthMax</code>	<code>WidthMax</code>
<code>GetHeightMax</code>	<code>HeightMax</code>
<code>GetDeviceVendorName</code>	<code>DeviceVendorName</code>
<code>GetDeviceModelName</code>	<code>GetDeviceModelName</code>
<code>GetDeviceManufacturerInfo</code>	<code>DeviceManufacturerInfo</code>
<code>GetDeviceVersion</code>	<code>DeviceVersion</code>
<code>GetDeviceFirmwareVersion</code>	<code>DeviceFirmwareVersion</code>
<code>GetDeviceID</code>	<code>DeviceID</code>
<code>PutDeviceUserID</code>	<code>DeviceUserID</code>
<code>GetDeviceUserID</code>	
<code>DeviceReset</code>	<code>DeviceReset</code>
<code>PutDeviceScanType</code>	<code>DeviceScanType</code>
<code>GetDeviceScanType</code>	
<code>PutWidth</code>	<code>Width</code>

Execute command	CBaslerGigEInstantCamera class member
GetWidth	
PutHeight	Height
GetHeight	
PutOffsetX	OffsetX
GetOffsetX	
PutOffsetY	OffsetY
GetOffsetY	
PutCenterX	CenterX
GetCenterX	
PutCenterY	CenterY
GetCenterY	
PutLineSelector	LineSelector
GetLineSelector	
PutLineInverter	LineInverter
GetLineInverter	
PutLineDebouncerTimeAbs	LineDebouncerTimeAbs
GetLineDebouncerTimeAbs	
GetLineStatus	LineStatus
GetLineStatusAll	
PutLineMode	LineMode
GetLineMode	
PutLineTermination	LineTermination
GetLineTermination	
PutLineSource	LineSource
GetLineSource	
PutUserOutputSelector	UserOutputSelector
GetUserOutputSelector	
PutUserOutputValue	UserOutputValue
GetUserOutputValue	
GetUserOutputValueAll	UserOutputValueAll
PutUserSetSelector	UserSetSelector
GetUserSetSelector	
PutExposureTimeRaw	ExposureTimeRaw
GetExposureTimeRaw	

