

Basler Pylon GigE プロバイダ

Version 1.2.0

ユーザーズ ガイド

June 30, 2020

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2013-07-17	初版.
1.0.1	2014-06-19	PutUserSetSelector, GetUserSetSelector, PutExposureTimeRaw, GetExposureTimeRaw, コマンド追加
1.1.0	2018-01-22	Windows 10 に対応
	2018-09-20	Pylon5 バージョン明記 5.0.12 対応
	2018-10-12	カメラのトリガーモードが OFF 時のエラーコードを追加
	2019-10-29	Pylon5 のセットアップ手順を追加
1.2.0	2020-06-01	Pylon SDK Version6 に対応

【対応機器】

機種	バージョン	注意事項
ace640-100gm		
aceA1300-30gg		

目次

1. はじめに	6
2. プロバイダの概要	7
2.1. 概要	7
2.2. メソッド・プロパティ	8
2.2.1. CaoWorkspace::AddController メソッド	8
2.2.2. CaoController::AddVariable メソッド	9
2.2.3. CaoController::Execute メソッド	9
2.2.4. CaoVariable::get_Value メソッド	9
2.2.5. CaoVariable::put_Value メソッド	9
2.3. 変数一覧	10
2.3.1. コントローラクラス	10
2.4. エラーコード	10
3. コマンドリファレンス	11
3.1. コントローラクラス	11
3.1.1. CaoController::Execute(“Start”) コマンド	12
3.1.2. CaoController::Execute(“Stop”) コマンド	12
3.1.3. CaoController::Execute(“GetImage”) コマンド	13
3.1.4. CaoController::Execute(“GetLastErrorDescription”) コマンド	13
3.1.5. CaoController::Execute(“ExecSoftTrigger”) コマンド	14
3.1.6. CaoController::Execute(“GetSensorWidth”) コマンド	14
3.1.7. CaoController::Execute(“GetSensorHeight”) コマンド	14
3.1.8. CaoController::Execute(“GetWidthMax”) コマンド	15
3.1.9. CaoController::Execute(“GetHeightMax”) コマンド	15
3.1.10. CaoController::Execute(“GetDeviceVendorName”) コマンド	15
3.1.11. CaoController::Execute(“GetDeviceModelName”) コマンド	16
3.1.12. CaoController::Execute(“GetDeviceManufacturerInfo”) コマンド	16
3.1.13. CaoController::Execute(“GetDeviceVersion”) コマンド	16
3.1.14. CaoController::Execute(“GetDeviceFirmwareVersion”) コマンド	17
3.1.15. CaoController::Execute(“GetDeviceID”) コマンド	17
3.1.16. CaoController::Execute(“PutDeviceUserID”) コマンド	17
3.1.17. CaoController::Execute(“GetDeviceUserID”) コマンド	18
3.1.18. CaoController::Execute(“DeviceReset”) コマンド	18

3.1.19. CaoController::Execute(“PutDeviceScanType”) コマンド	18
3.1.20. CaoController::Execute(“GetDeviceScanType”) コマンド	19
3.1.21. CaoController::Execute(“PutWidth”) コマンド	19
3.1.22. CaoController::Execute(“GetWidth”) コマンド	19
3.1.23. CaoController::Execute(“PutHeight”) コマンド	20
3.1.24. CaoController::Execute(“GetHeight”) コマンド	20
3.1.25. CaoController::Execute(“PutOffsetX”) コマンド	20
3.1.26. CaoController::Execute(“GetOffsetX”) コマンド	21
3.1.27. CaoController::Execute(“PutOffsetY”) コマンド	21
3.1.28. CaoController::Execute(“GetOffsetY”) コマンド	21
3.1.29. CaoController::Execute(“PutCenterX”) コマンド	22
3.1.30. CaoController::Execute(“GetCenterX”) コマンド	22
3.1.31. CaoController::Execute(“PutCenterY”) コマンド	22
3.1.32. CaoController::Execute(“GetCenterY”) コマンド	23
3.1.33. CaoController::Execute(“PutLineSelector”) コマンド	23
3.1.34. CaoController::Execute(“GetLineSelector”) コマンド	24
3.1.35. CaoController::Execute(“PutLineInverter”) コマンド	24
3.1.36. CaoController::Execute(“GetLineInverter”) コマンド	24
3.1.37. CaoController::Execute(“PutLineDebouncerTimeAbs”) コマンド	25
3.1.38. CaoController::Execute(“GetLineDebouncerTimeAbs”) コマンド	25
3.1.39. CaoController::Execute(“GetLineStatus”) コマンド	25
3.1.40. CaoController::Execute(“GetLineStatusAll”) コマンド	26
3.1.41. CaoController::Execute(“PutLineMode”) コマンド	26
3.1.42. CaoController::Execute(“GetLineMode”) コマンド	26
3.1.43. CaoController::Execute(“PutLineTermination”) コマンド	27
3.1.44. CaoController::Execute(“GetLineTermination”) コマンド	27
3.1.45. CaoController::Execute(“PutLineSource”) コマンド	27
3.1.46. CaoController::Execute(“GetLineSource”) コマンド	28
3.1.47. CaoController::Execute(“PutUserOutputSelector”) コマンド	28
3.1.48. CaoController::Execute(“GetUserOutputSelector”) コマンド	29
3.1.49. CaoController::Execute(“PutUserOutputValue”) コマンド	29
3.1.50. CaoController::Execute(“GetUserOutputValue”) コマンド	29
3.1.51. CaoController::Execute(“GetUserOutputValueAll”) コマンド	30
3.1.52. CaoController::Execute(“PutUserSetSelector”) コマンド	30
3.1.53. CaoController::Execute(“GetUserSetSelector”) コマンド	31
3.1.54. CaoController::Execute(“PutExposureTimeRaw”) コマンド	32
3.1.55. CaoController::Execute(“GetExposureTimeRaw”) コマンド	32

付録 A. Pylon 5(SDK Version 5.0.12)セットアップ手順.....	33
付録 B. Pylon6(SDK Version 6.1.0) セットアップ手順	38
付録 C. Pylon API 対応表	41

1. はじめに

本書は CAO プロバイダの一つである, PylonGigE プロバイダのユーザーズガイドです.

PylonGigE プロバイダは, Basler 製カメラコントロールソフトウェア Pylon を使用して GigE Vision カメラ(以下 GigE カメラ)と通信を行います.

PylonGigE プロバイダを使用するためには, 以下の Basler 製カメラコントロールソフトウェアのいずれかが必要です.

[Pylon SDK Version 4.0.0](#)

[Pylon SDK Version 5.0.12](#)

[Pylon SDK Version 6.1.0](#)

なお, Versoin4 系は Windows 10 OS にはインストールできませんのでご注意ください.

次章に PylonGigE プロバイダの概要, 3 章にコマンドリファレンスを記載しています.

2. プロバイダの概要

2.1. 概要

PylonGigE プロバイダは, Pylon API に依存する部分を吸収し CAO プロバイダ・インターフェース仕様で規定された機能を提供する CAO プロバイダです. GigE カメラとの通信は Pylon の CBaslerGigEInstantCamera クラスを使用しています.

PylonGigE プロバイダは起動時に GigE カメラのデフォルトスタートアップ設定をロードします. デフォルトスタートアップの設定方法については, Basler 製カメラコントロールソフトウェア Pylon の“C++ Programmer's Guide and Reference Documentation”を参照してください.

そのファイル形式は DLL(Dynamic Link Library)であり, CAO エンジンから使用時に動的にロードされます. PylonGigE プロバイダを使用するにあたっては ORiN2SDK をインストールするか, 下表を参照して手作業でレジストリ登録を行う必要があります.

表 2-1 PylonGigE プロバイダ

ファイル名	CaoProvPylonGigE.dll
ProgID	CaoProv.Basler.PylonGigE
レジストリ登録	regsvr32 CaoProvPylonGigE.dll
レジストリ登録の抹消	regsvr32 /u CaoProvPylonGigE.dll

なお, PylonGigE プロバイダは, 使用する PC のインストール状況によって, 別途プロバイダのレジストリ登録が必要になります.

表 2-2 追加プロバイダのレジストリ登録

Pylon SDK Version 4.0.0 をインストールしている環境	regsvr32 CaoProvPylon4GigE.dll
Pylon SDK Version 5.0.12 をインストールしている環境	regsvr32 CaoProvPylon5GigE.dll
Pylon SDK Version 6.1.0 をインストールしている環境	regsvr32 CaoProvPylon6GigE.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

PylonGigE プロバイダでは AddController 時に GigE カメラとの接続を行います。
このときオプションで接続先のカメラを指定します。



AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>)

<bstrCtrlName> : [in] ロボット名
 <bstrProvName> : [in] オプション文字列
 固定値 =” CaoProv.Basler.PylonGigE”
 <bstrPcName> : [in] プロバイダの実行マシン名
 <bstrOption> : [in] オプション文字列

以下にオプション文字列に指定するリストを示す。

表 2-3 CaoWorkspace::AddController のオプション文字列

オプション	意味
Conn=Eth:<IP アドレス>	接続先カメラの IP アドレスを指定します。
Serial=<シリアル番号>	接続先カメラのシリアル番号を指定します。 Conn オプションが同時に指定されていた場合、このオプションは無視されます。
UserID=<ユーザ ID>	接続先カメラのユーザ ID を指定します。 Conn オプション又は Serial オプションが指定されていた場合、このオプションは無視されます。
ManualStart[=<True/False>]	GigE カメラ接続後、画像取得の開始を手動で行うかどうかを指定します。(デフォルト:False) 手動で行う場合は CaoController::Execute()の“Start”コマンドを実行してください。
Timeout[=<タイムアウト時間>]	画像取得時の待機時間を指定します。 (デフォルト:1000ms)
Version=<対象バージョン>	どのバージョンの Pylon SDK がインストールされているかを指定します。 入力できる値は 4 / 5 / 6 です。 また、何も指定しない場合はプロバイダのレジストリパラメータにしている値となります。

接続先を指定するオプションには優先順位があります。優先度が高いオプションが使用された場合、優先度の低いオプションは無視されます。オプションの優先度は以下のようになっています。

Conn オプション > Serial オプション > UserID オプション

2.2.2. GaoController::AddVariable メソッド

変数オブジェクトを生成します。

PylonGigE プロバイダで実装されている変数は 2.3.1 を参照してください。

書式

AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

<bstrVariableName > : [in] 変数名

<bstrOption> : [in] オプション文字列

使用例

```
Dim aaa As Object
Dim bbb As Double

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

2.2.3. GaoController::Execute メソッド

コマンドを実行します。

Execute メソッドの引数は、コマンドを BSTR、パラメータを VARIANT 配列で指定します。

各コマンドの詳細は 3.1 を参照してください。

書式

[<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])

<vntRet > : [out] コマンドの戻り値

<bstrCmd > : [in] コマンド

<vntParam > : [in] パラメータ

2.2.4. GaoVariable::get_Value メソッド

変数の値を取得します。

取得する値の詳細については 2.3 を参照してください。

2.2.5. GaoVariable::put_Value メソッド

変数の値を設定します。

設定する値の詳細については 2.3 を参照してください。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-4 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@IMAGE	VT_I4	画像の取得. カメラのトリガーモードが ON の場合は値取得前に ExecSoftTrigger コマンドを実行してください.	○	-

2.4. エラーコード

PylonGigE プロバイダでは、以下の固有エラーコードが定義されています。ORiN2 共通エラーについては、[「ORiN2 プログラミングガイド」](#)のエラーコードの章を参照してください。

表 2-5 独自エラーコード一覧

エラー名	エラー番号	説明
E_CAOP_PYLON_ERR	0x80100001	Pylon API でエラーが発生しました。 CaoController::Execute() の “GetLastErrorDescription”で詳細を取得してください。
E_CAOP_UNINIT	0x80100002	未初期化状態です。
E_CAOP_NOIMAGE	0x80100003	画像がありません。
E_CAOP_DEVICE_NOTFOUND	0x80100004	カメラデバイスが見つかりません。
E_CAOP_DEVICE_INVALID	0x80100005	カメラデバイスの指定が間違っています。
E_CAOP_TRIGGERMODE_OFF	0x80100006	カメラのトリガーモードが OFF になっています。
E_CAOP_VERSION_INVALID	0x80100007	バージョン指定が間違っています。

3. コマンドリファレンス

本章では CaoController::Execute メソッドの各コマンドについて解説します。

CaoController::Execute メソッドの各コマンドは、Pylon API の CBaslerGigEInstantCamera クラスメンバを実行します。各コマンドと PylonAPI との対応については付録 A を参照してください。

詳細動作については、Basler 製カメラコントロールソフトウェア Pylon の“C++ Programmer's Guide and Reference Documentation”を参照してください。

3.1. コントローラクラス

表 3-1 CaoController::Execute コマンド一覧

コマンド	機能	
Start	画像取込みの開始	P. 12
Stop	画像取込みの停止	P. 12
GetImage	最新画像の取得	P. 13
GetLastErrorDescription	Pylon API エラー詳細の取得	P. 13
ExecSoftTrigger	ソフトウェアトリガの実行	P. 14
GetSensorWidth	カメラセンサの幅の取得	P. 14
GetSensorHeight	カメラセンサの高さの取得	P. 14
GetWidthMax	最大の幅の取得	P. 15
GetHeightMax	最大の高さの取得	P. 15
GetDeviceVendorName	ベンダ名の取得	P. 15
GetDeviceModelName	モデル名の取得	P. 16
GetDeviceManufacturerInfo	製造元からの追加情報の取得	P. 16
GetDeviceVersion	バージョンの取得	P. 16
GetDeviceFirmwareVersion	ファームウェアバージョンの取得	P. 17
GetDeviceID	デバイス ID の取得	P. 17
PutDeviceUserID	ユーザ ID の設定	P. 17
GetDeviceUserID	ユーザ ID の取得	P. 18
DeviceReset	デバイスのリセット	P. 18
PutDeviceScanType	デバイスセンサのスキャンタイプの設定	P. 18
GetDeviceScanType	デバイスセンサのスキャンタイプの取得	P. 19
PutWidth	AOI の幅の設定	P. 19
GetWidth	AOI の幅の取得	P. 19
PutHeight	AOI の高さの設定	P. 20
GetHeight	AOI の高さの取得	P. 20
PutOffsetX	AOI の X オフセットの設定	P. 20
GetOffsetX	AOI の X オフセットの取得	P. 21
PutOffsetY	AOI の Y オフセットの設定	P. 21
GetOffsetY	AOI の Y オフセットの取得	P. 21

PutCenterX	水平方向センタリングフラグの設定	P. 22
GetCenterX	水平方向センタリングフラグの取得	P. 22
PutCenterY	垂直方向センタリングフラグの設定	P. 22
GetCenterY	垂直方向センタリングフラグの取得	P. 23
PutLineSelector	選択中 I/O の切り替え	P. 23
GetLineSelector	選択中 I/O の取得	P. 24
PutLineInverter	選択中 I/O の信号インバータ機能の設定	P. 24
GetLineInverter	選択中 I/O の信号インバータ機能の取得	P. 24
PutLineDebouncerTimeAbs	選択中 I/O のデバウンサー時間に対する絶対値の設定	P. 25
GetLineDebouncerTimeAbs	選択中 I/O のデバウンサー時間に対する絶対値の取得	P. 25
GetLineStatus	選択中 I/O の論理状態の取得	P. 25
GetLineStatusAll	全ての I/O の論理状態の取得	P. 26
PutLineMode	選択中 I/O のモードの設定	P. 26
GetLineMode	選択中 I/O のモードの取得	P. 26
PutLineTermination	選択中 IN に終端抵抗機能の設定	P. 27
GetLineTermination	選択中 IN に終端抵抗機能の取得	P. 27
PutLineSource	選択中 OUT に対してソース信号の設定	P. 27
GetLineSource	選択中 OUT に対するソース信号の取得	P. 28
PutUserOutputSelector	選択中ユーザ出力の設定	P. 28
GetUserOutputSelector	選択中ユーザ出力の取得	P. 29
PutUserOutputValue	選択中ユーザ出力の値の設定	P. 29
GetUserOutputValue	選択中ユーザ出力の値の出力	P. 29
GetUserOutputValueAll	全てのユーザ出力の値の取得	P. 30
PutUserSetSelector	選択中のユーザーセット番号設定	P. 30
GetUserSetSelector	選択中のユーザーセット番号取得	P. 31
PutExposureTimeRaw	露光時間の設定	P. 32
GetExposureTimeRaw	露光時間の取得	P. 32

3.1.1. GaoController::Execute(“Start”) コマンド

GigE カメラによる画像取込みを開始します。

書式 Start()

戻り値 : なし

使用例

```
caoCtrl.Start ' 画像取込み開始
```

3.1.2. GaoController::Execute(“Stop”) コマンド

GigE カメラによる画像取込みを停止します。

取り込み済みの画像はクリアされます。

書式 Stop()

戻り値 : なし

使用例

```
caoCtrl.Stop ' 画像取り込み停止'
```

3.1.3. CaoController::Execute(“GetImage”) コマンド

GigE カメラによって取り込まれた画像を取得します。

画像が取り込まれていない場合は, AddController() の Timeout オプションで指定した時間の間, 画像が来るのを待ちます。

カメラのトリガーモードが ON の場合は本コマンド実行前に ExecSoftTrigger コマンドを実行してください。このコマンドが成功すると取り込み済みの画像はクリアされます。

書式 GetImage()

戻り値 : [out] 画像 (VT_UI1 | VT_ARRAY)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetImage '画像取得'
```

3.1.4. CaoController::Execute(“GetLastErrorDescription”) コマンド

最後に発生した Pylon API のエラー詳細を取得します。

Pylon API のエラーが発生した場合, E_CAOP_PYLON_ERR (0x80100001)が発生します。この直後にこのコマンドを実行することで発生したエラーの詳細を確認することができます。

書式 GetLastErrorDescription ()

戻り値 : [out] エラーの詳細 (VT_BSTR)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetLastErrorDescription 'エラー詳細の取得
```

3.1.5. CaoController::Execute(“ExecSoftTrigger”) コマンド

GigE カメラに対してソフトウェアトリガを実行します。カメラのトリガーモードが OFF の場合は本コマンドも実行しても何も起こりません。

書式 ExecSoftTrigger()

戻り値 : なし

使用例

```
caoCtrl.ExecSoftTrigger 'ソフトウェアトリガ実行
```

3.1.6. CaoController::Execute(“GetSensorWidth”) コマンド

GigE カメラのセンサの幅を取得します

書式 GetSensorWidth()

戻り値 : [out] 幅 (VT_I4)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetSensorWidth 'センサ幅の取得
```

3.1.7. CaoController::Execute(“GetSensorHeight”) コマンド

GigE カメラのセンサの高さを取得します

書式 GetSensorHeight()

戻り値 : [out] 高さ (VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetSensorHeight 'センサ高の取得
```

3.1.8. CaoController::Execute(“GetWidthMax”) コマンド

GigE カメラの幅の最大値を取得します

書式 GetWidthMax()

戻り値 : [out] 幅 (VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetWidthMax '幅の最大値の取得
```

3.1.9. CaoController::Execute(“GetHeightMax”) コマンド

GigE カメラのセンサの高さの最大値を取得します

書式 GetHeightMax()

戻り値 : [out] 高さ (VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetHeightMax '高さの最大値の取得
```

3.1.10. CaoController::Execute(“GetDeviceVendorName”) コマンド

GigE カメラのベンダ名を取得します

書式 GetDeviceVendorName()

戻り値 : [out] ベンダ名 (VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceVendorName 'ベンダ名の取得
```

3.1.11. CaoController::Execute(“GetDeviceModelName”) コマンド

GigE カメラのモデル名を取得します

書式 GetDeviceModelName()

戻り値 : [out] モデル名 (VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceModelName 'モデル名の取得
```

3.1.12. CaoController::Execute(“GetDeviceManufacturerInfo”) コマンド

GigE カメラの製造元からの追加情報を取得します

書式 GetDeviceManufacturerInfo()

戻り値 : [out] 追加情報 (VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceManufacturerInfo '追加情報の取得
```

3.1.13. CaoController::Execute(“GetDeviceVersion”) コマンド

GigE カメラのデバイスバージョンを取得します

書式 GetDeviceVersion()

戻り値 : [out] デバイスバージョン (VT_BSTR)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetDeviceVersion 'デバイスバージョンの取得
```

3.1.14. CaoController::Execute("GetDeviceFirmwareVersion") コマンド

GigE カメラのファームウェアバージョンを取得します

書式 GetDeviceFirmwareVersion()

戻り値 : [out] ファームウェアバージョン (VT_BSTR)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetDeviceFirmwareVersion 'ファームウェアバージョンの取得
```

3.1.15. CaoController::Execute("GetDeviceID") コマンド

GigE カメラのデバイス ID を取得します

書式 GetDeviceID()

戻り値 : [out] デバイス ID (VT_BSTR)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetDeviceID 'デバイス ID の取得
```

3.1.16. CaoController::Execute("PutDeviceUserID") コマンド

GigE カメラのユーザ ID を設定します

書式 PutDeviceUserID(<bstrUserID:VT_BSTR>)

<bstrUserID> : [in] ユーザ ID(VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutDeviceUserID("Sample") 'ユーザ ID の設定
```

3.1.17. CaoController::Execute("GetDeviceUserID") コマンド

GigE カメラのユーザ ID を取得します

書式 GetDeviceUserID()

戻り値 : [out] ユーザ ID (VT_BSTR)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetDeviceUserID 'デバイス ID の取得
```

3.1.18. CaoController::Execute("DeviceReset") コマンド

GigE カメラのリセットを実行します。

このコマンドを実行すると、GigE カメラと切断されます。GigE カメラとの再接続を行う場合は、現在のCaoController オブジェクトを破棄し、AddController()を再度実行してください。

書式 DeviceReset()

戻り値 : なし

使用例

```
caoCtrl.DeviceReset      ' リセット実行
```

3.1.19. CaoController::Execute("PutDeviceScanType") コマンド

GigE カメラのデバイスセンサのスキャンタイプを設定します

書式 PutDeviceScanType(<lScanType:VT_BSTR>)

<lScanType> : [in] スキャンタイプ(VT_I4)

PylonAPI の Basler_GigECamera::DeviceScanTypeEnums の値

を使用します。

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.PutDeviceUserID(0) 'スキャンタイプの設定
```

3.1.20. CaoController::Execute("GetDeviceScanType") コマンド

GigE カメラのデバイスセンサのスキャンタイプを取得します

書式 GetDeviceScanType()

戻り値 : [out] スキャンタイプ(VT_I4)

PylonAPI の Basler_GigECamera::DeviceScanTypeEnums の値
を使用します。

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetDeviceScanType 'スキャンタイプの取得
```

3.1.21. CaoController::Execute("PutWidth") コマンド

AOI の幅を設定します

書式 PutWidth(<IWidth:VT_BSTR>)

<IScanType> : [in] 幅(VT_I4)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.PutWidth(320) '幅の設定
```

3.1.22. CaoController::Execute("GetWidth") コマンド

AOI の幅を取得します

書式 GetWidth()

戻り値 : [out] 幅(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetWidth '幅の取得
```

3.1.23. CaoController::Execute(“PutHeight”) コマンド

AOI の高さを設定します

書式 PutHeight(<IHeight:VT_BSTR>)

<IScanType> : [in] 高さ(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutHeight(240) '高さの設定
```

3.1.24. CaoController::Execute(“GetHeight”) コマンド

AOI の高さを取得します

書式 GetHeight()

戻り値 : [out] 高さ(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetHeight '高さの取得
```

3.1.25. CaoController::Execute(“PutOffsetX”) コマンド

AOI の X オフセットを設定します

書式 PutOffsetX(<IOffsetX:VT_BSTR>)

<IOffsetX> : [in] X オフセット(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl. PutOffsetX(100) 'X オフセットの設定
```

3.1.26. CaoController::Execute(“GetOffsetX”) コマンド

AOI の X オフセットを取得します

書式 GetOffsetX()

戻り値 : [out] X オフセット(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl. GetOffsetX 'X オフセットの取得
```

3.1.27. CaoController::Execute(“PutOffsetY”) コマンド

AOI の Y オフセットを設定します

書式 PutOffsetY(<IOffsetY:VT_BSTR>)

<IOffsetY> : [in] Y オフセット(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl. PutOffsetY(100) 'Y オフセットの設定
```

3.1.28. CaoController::Execute(“GetOffsetY”) コマンド

AOI の Y オフセットを取得します

書式 GetOffsetY()

戻り値 : [out] Y オフセット(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetOffsetY 'Y オフセットの取得
```

3.1.29. CaoController::Execute("PutCenterX") コマンド

AOI の水平方向センタリングフラグを設定します

書式 PutCenterX(<lCenterX:VT_BSTR>)

<lCenterX> : [in] センタリングフラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutCenterX(100) 'センタリングフラグの設定
```

3.1.30. CaoController::Execute("GetCenterX") コマンド

AOI の水平方向センタリングフラグを取得します

書式 GetCenterX()

戻り値 : [out] センタリングフラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetCenterX 'センタリングフラグの取得
```

3.1.31. CaoController::Execute("PutCenterY") コマンド

AOI の垂直方向センタリングフラグを設定します

書式

PutCenterY(<lCenterY:VT_BSTR>)

<lCenterY> : [in] センタリングフラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutCenterY(100) 'センタリングフラグの設定
```

3.1.32. CaoController::Execute(“GetCenterY”) コマンド

AOI の垂直方向センタリングフラグを取得します

書式

GetCenterY()

戻り値 : [out] センタリングフラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetCenterY 'センタリングフラグの取得
```

3.1.33. CaoController::Execute(“PutLineSelector”) コマンド

選択中の I/O を切り替えます。

書式

PutLineSelector(<lIO:VT_I4>)

<lIO> : [in] 選択 I/O(VT_I4)

PylonAPI の Basler_GigECamera::LineSelectorEnums の値を使用します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineSelector(0) 'I/O 切り替え
```

3.1.34. CaoController::Execute(“GetLineSelector”) コマンド

選択中の I/O を取得します。

書式 GetLineSelector()

戻り値 : [out] 選択 I/O(VT_I4)

PylonAPI の Basler_GigECamera::LineSelectorEnums の値を使用します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineSelector '選択中 I/O の取得
```

3.1.35. CaoController::Execute(“PutLineInverter”) コマンド

選択中 I/O の信号インバータ機能の ON/OFF を設定します。

書式 PutLineInverter(<IEnable:VT_BOOL>)

<IEnable> : [in] 信号インバータ機能フラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineInverter(false) '信号インバータ機能設定
```

3.1.36. CaoController::Execute(“GetLineInverter”) コマンド

選択中 I/O の信号インバータ機能の ON/OFF 状態を取得します。

書式 GetLineInverter()

戻り値 : [out] 信号インバータ機能フラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineInverter '信号インバータ機能状態取得
```

3.1.37. CaoController::Execute(“PutLineDebouncerTimeAbs”) コマンド

選択中 I/O のデバウンサー時間の絶対値を設定します。(単位:マイクロ秒)

書式 PutLineDebouncerTimeAbs(<dTime:VT_BOOL>)

<dTime> : [in] デバウンサー時間(VT_R8)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineDebouncerTimeAbs(10.00) 'デバウンサー時間設定
```

3.1.38. CaoController::Execute(“GetLineDebouncerTimeAbs”) コマンド

選択中 I/O のデバウンサー時間の絶対値を取得します。(単位:マイクロ秒)

書式 GetLineDebouncerTimeAbs()

戻り値 : [out] デバウンサー時間(VT_R8)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineDebouncerTimeAbs 'デバウンサー時間取得
```

3.1.39. CaoController::Execute(“GetLineStatus”) コマンド

選択中 I/O の論理状態を取得します。

書式 GetLineStatus()

戻り値 : [out] 状態(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineStatus '状態取得
```

3.1.40. CaoController::Execute(“GetLineStatusAll”) コマンド

全ての I/O の論理状態を取得します。

書式 GetLineStatusAll()

戻り値 : [out] 状態(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineStatusAll    '状態取得
```

3.1.41. CaoController::Execute(“PutLineMode”) コマンド

選択中 I/O のモードを設定します。

書式 PutLineMode(<IMode:VT_I4>)

<IMode> : [in] モード(VT_I4)

PylonAPI の Basler_GigECamera::LineModeEnums の値を使用
します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineMode(0)    'モード設定
```

3.1.42. CaoController::Execute(“GetLineMode”) コマンド

選択中 I/O のモードを取得します。

書式 GetLineMode()

戻り値 : [out] モード(VT_I4)

PylonAPI の Basler_GigECamera::LineModeEnums の値を使用
します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineMode 'モード取得
```

3.1.43. CaoController::Execute(“PutLineTermination”) コマンド

選択中である IN の終端抵抗機能の ON/OFF を設定します。

書式

PutLineTermination(<lEnable:VT_BOOL>)

<lEnable> : [in] 終端抵抗機能フラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineTermination(false) '終端抵抗機能設定
```

3.1.44. CaoController::Execute(“GetLineTermination”) コマンド

選択中である IN の終端抵抗機能の ON/OFF 状態を取得します。

書式

GetLineTermination()

戻り値 : [out] 終端抵抗機能フラグ(VT_BOOL)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineTermination '終端抵抗機能状態取得
```

3.1.45. CaoController::Execute(“PutLineSource”) コマンド

選択中である OUT のソース信号を設定します。

書式

PutLineSource(<lSource:VT_I4>)

<lSource> : [in] ソース信号(VT_I4)

PylonAPI の Basler_GigECamera::LineSourceEnums の値を使用します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutLineSource(0) 'モード設定
```

3.1.46. CaoController::Execute(“GetLineSource”) コマンド

選択中である OUT のソース信号を取得します。

書式

GetLineSource()

戻り値 : [out] ソース信号(VT_I4)

PylonAPI の Basler_GigECamera::LineSourceEnums の値を使用します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetLineSource 'モード取得
```

3.1.47. CaoController::Execute(“PutUserOutputSelector”) コマンド

選択中のユーザ出力を切り替えます。

書式

PutUserOutputSelector(<IUserOutput:VT_I4>)

<IUserOutput> : [in] 選択ユーザ出力(VT_I4)

PylonAPI の Basler_GigECamera::SyncUserOutputSelectorEnums の値を使用します。

使用例

```
Dim vRes As Variant
vRes = caoCtrl.PutUserOutputSelector(0) 'ユーザ出力切り替え
```

3.1.48. CaoController::Execute(“GetUserOutputSelector”) コマンド

選択中のユーザ出力を取得します。

書式 GetUserOutputSelector()

戻り値 : [out] ユーザ出力(VT_I4)
 PylonAPI の Basler_GigECamera::SyncUserOutputSelectorEnums
 の値を使用します。

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetUserOutputSelector 'ユーザ出力の取得
```

3.1.49. CaoController::Execute(“PutUserOutputValue”) コマンド

選択中であるユーザ出力の値を設定します。

書式 PutUserOutputValue(<bVal:VT_BOOL>)

<bVal> : [in] 値(VT_BOOL)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.PutUserOutputValue(false) 'ユーザ出力値設定
```

3.1.50. CaoController::Execute(“GetUserOutputValue”) コマンド

選択中であるユーザ出力の値を取得します。

書式 GetUserOutputValue()

戻り値 : [out] 値(VT_BOOL)

使用例

```
Dim vRes As Variant  
vRes = caoCtrl.GetUserOutputValue 'ユーザ出力値取得
```

3.1.51. CgoController::Execute(“GetUserOutputValueAll”) コマンド

全てのユーザ出力の値を取得します。

書式 GetUserOutputValueAll()

戻り値 : [out] 値(VT_I4)

使用例

```
Dim vRes As Variant
vRes = caoCtrl.GetUserOutputValueAll 'ユーザ出力値取得
```

3.1.52. CgoController::Execute(“PutUserSetSelector”) コマンド

使用するパラメータセットの番号を変更します。

書式 PutUserSetSelector(<lParam>)

<lParam> : [in] ユーザセット番号(VT_I4)

Pylon SDK Version 4 の場合

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

Pylon SDK Version 5, 6 の場合

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 のみ)

使用例

```
caoCtrl.PutUserSetSelector 2 ' ユーザーセット 2 に変更する
```

3.1.53. CaoController::Execute(“GetUserSetSelector”) コマンド

現在使用しているパラメータセットの番号を取得します。

書式

GetUserSetSelector()

戻り値 : [out] パラメータセット番号(VT_I4)

Pylon SDK Version 4 の場合

- 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

Pylon SDK Version 5, 6 の場合

- 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 のみ)

使用例

```
Dim lRet As Long
lRet = caoCtrl.GetUserSetSelector ' パラメータセット番号の取得
```

3.1.54. CaoController::Execute(“PutExposureTimeRaw”) コマンド

露光時間パラメータを設定します。

書式 PutExposureTimeRaw(<lParam>)

<lParam> : [in] 値(VT_I4)

使用例

```
caoCtrl.PutExposureTimeRaw 1000 ' 露光時間パラメータを 1000 に設定
```

3.1.55. CaoController::Execute(“GetExposureTimeRaw”) コマンド

現在設定されている露光時間パラメータを取得します。

書式 GetExposureTimeRaw()

戻り値 : [out] 値(VT_I4)

使用例

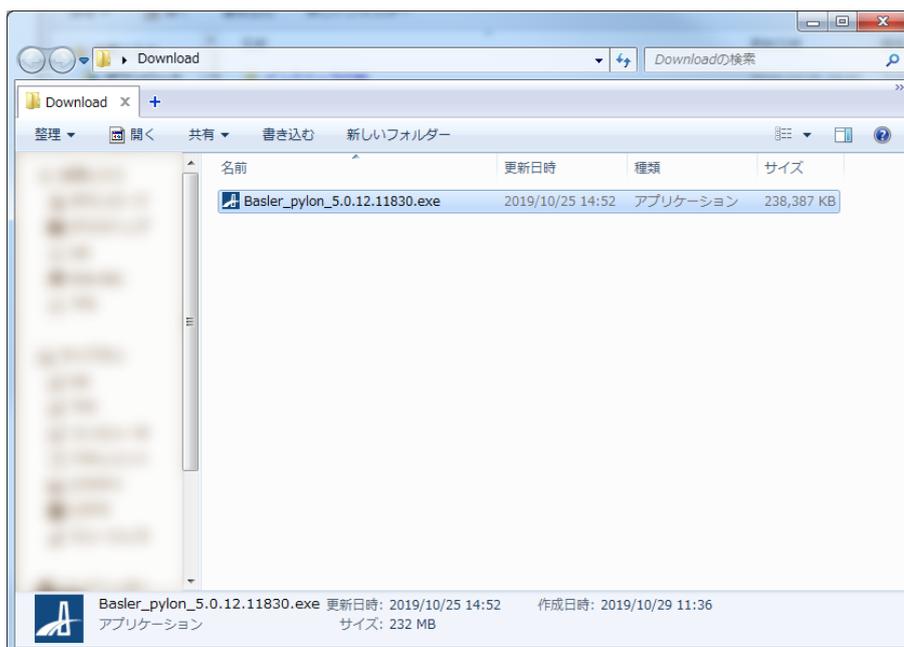
```
Dim lRet as Long
lRet = caoCtrl.GetExposureTimeRaw ' 露光時間パラメータ取得
```

付録A. Pylon 5(SDK Version 5.0.12)セットアップ手順

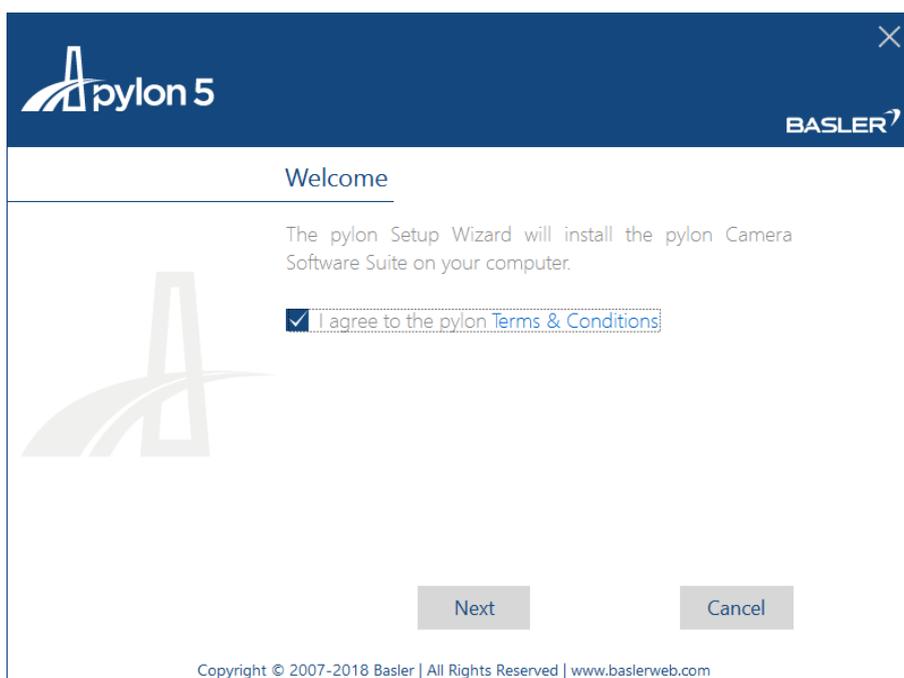
Pylon SDK Version 5.0.x の最新バージョンである 5.0.12 のセットアップ手順を記述します。

1. SDK のインストール

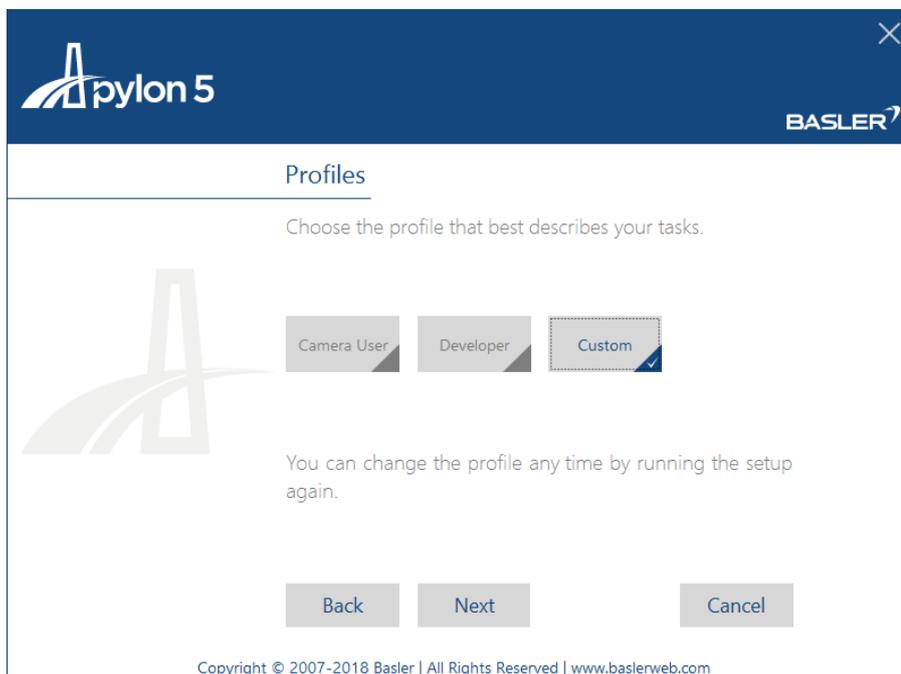
- Basler 社より Windows 用ソフトウェアスイート Pylon 5.0.12 をダウンロードしてください。
- ダウンロードした”Basler_pylon_5.0.12.11830.exe”を実行してください。



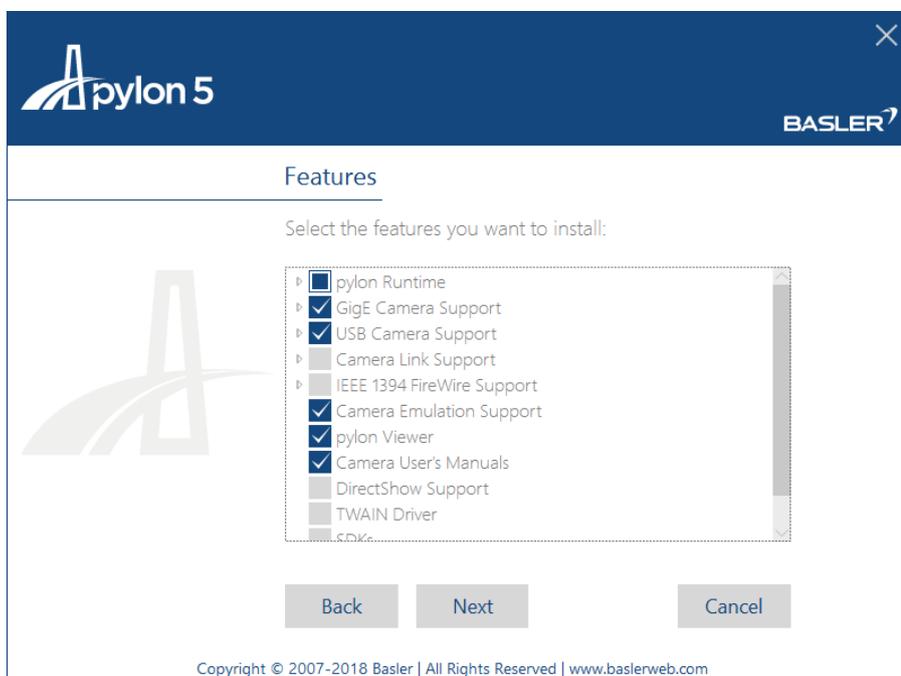
- 同意にチェックを入れ次へ進んでください。



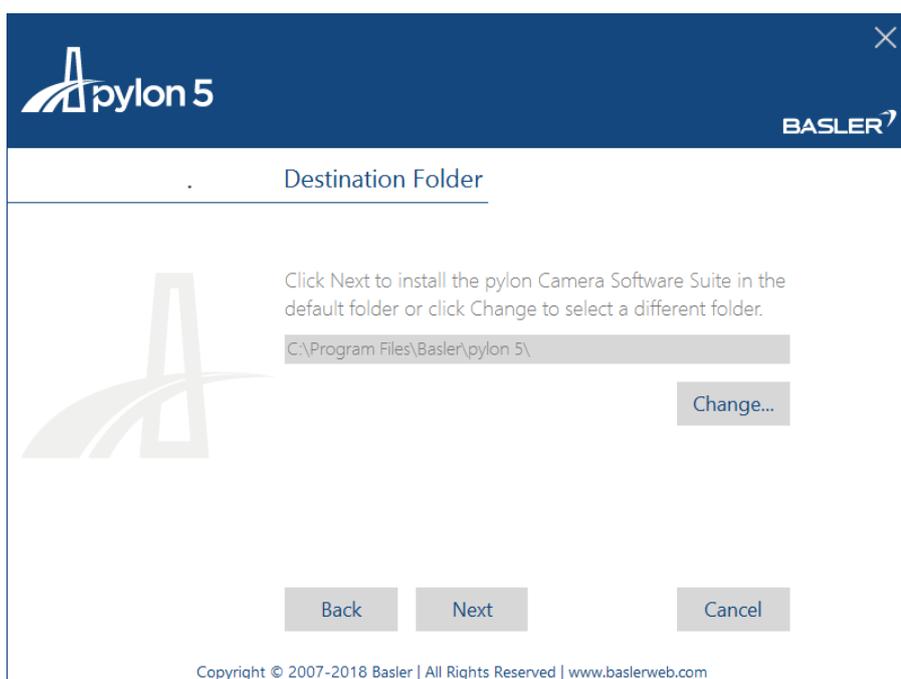
- ”Custom”を選択し次へ進んでください。



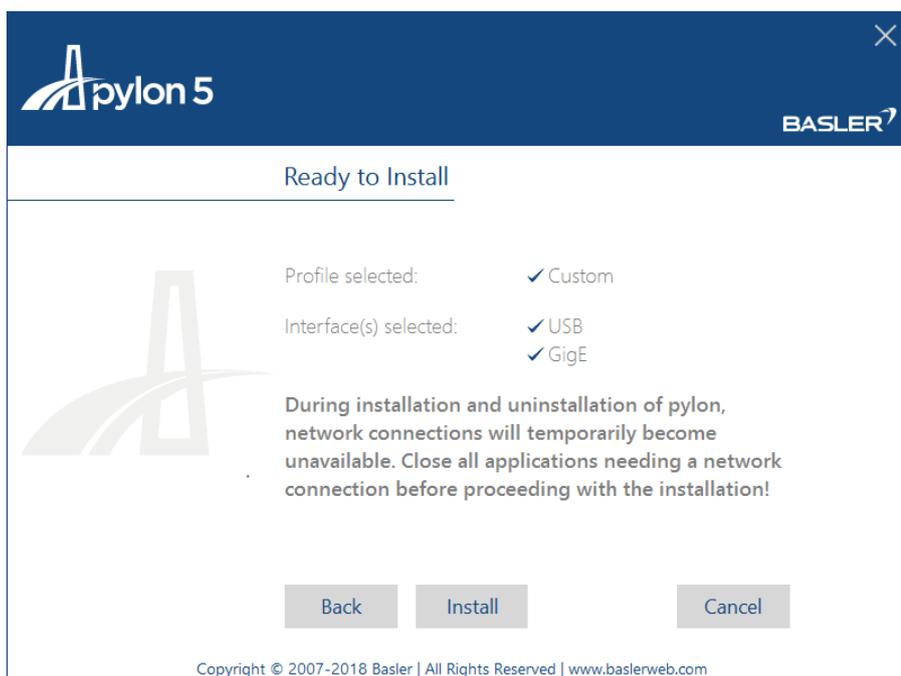
- デフォルトでチェックされている機能の中から “DirectShow Support”, “TWAIN Driver”, “Firmware Updater”のチェックを外し、次へ進んでください。



- 必要であればインストール先ディレクトリを変更し、次へ進んでください。

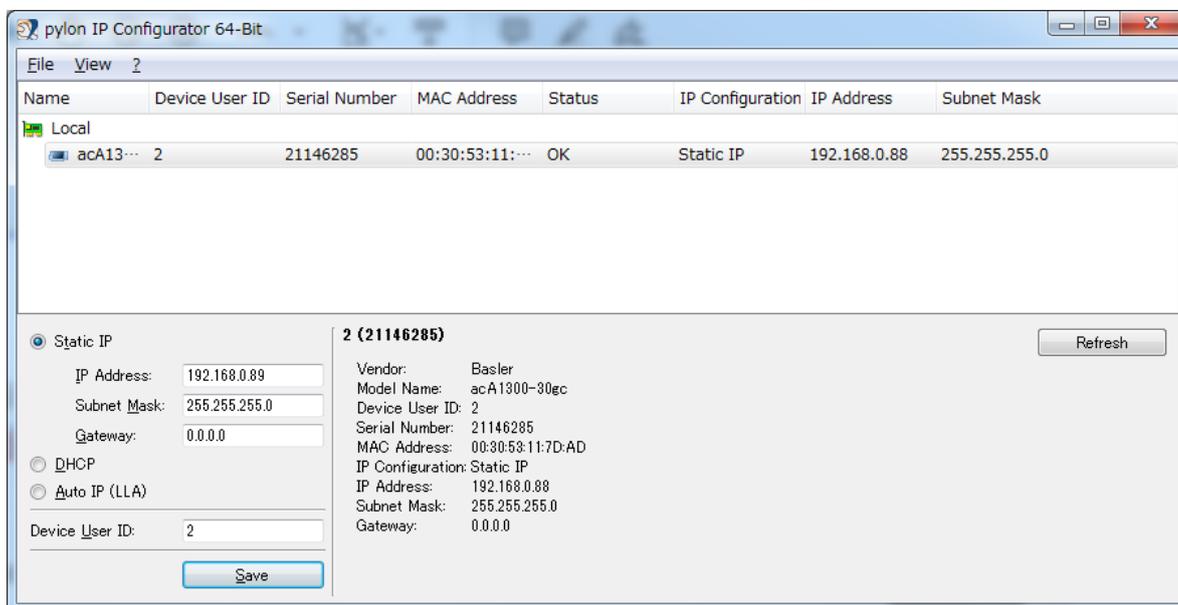


- インストールを実行してください。



2. GigE カメラの IP アドレスと ID の設定

- インストールされた pylon IP Configurator を起動し、設定したいカメラの“IP Address”, “ Subnet Mask”, “Gateway”, “Device User ID”を入力し, “Save”ボタンを押します.



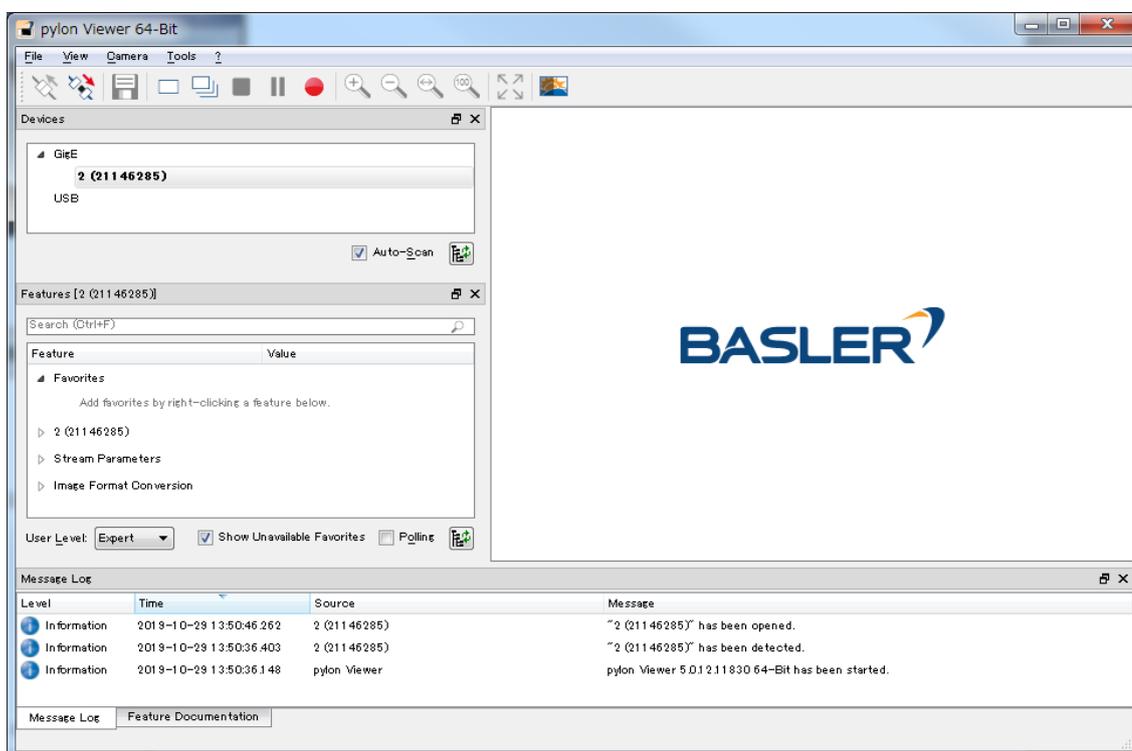
Device User ID の範囲は 0 – 4 です.

User ID 0 – 4 はコントローラのイメージ格納領域の ID201 – 205 に対応しています.

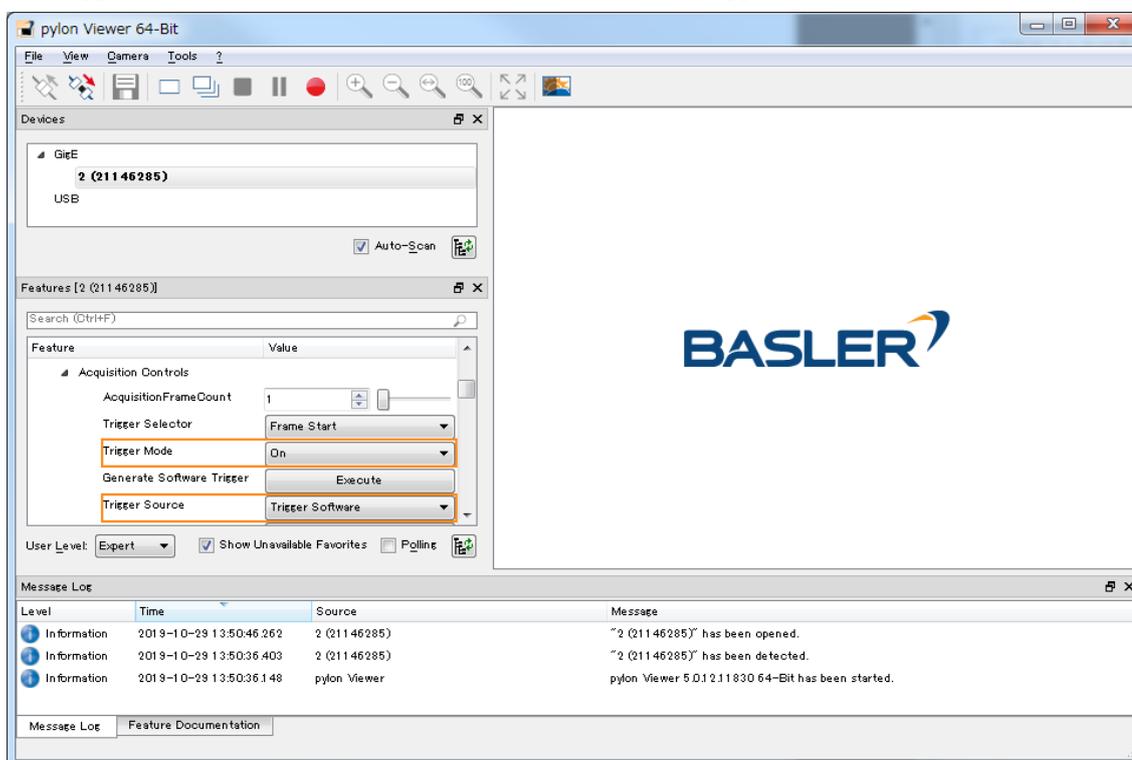
カメラを複数台使用するときは ID を重複しないようにしてください.

3. カメラの動作設定

- Pylon Viewer を起動し、カメラを選択し、“User Level”を “Expert”以上にします。

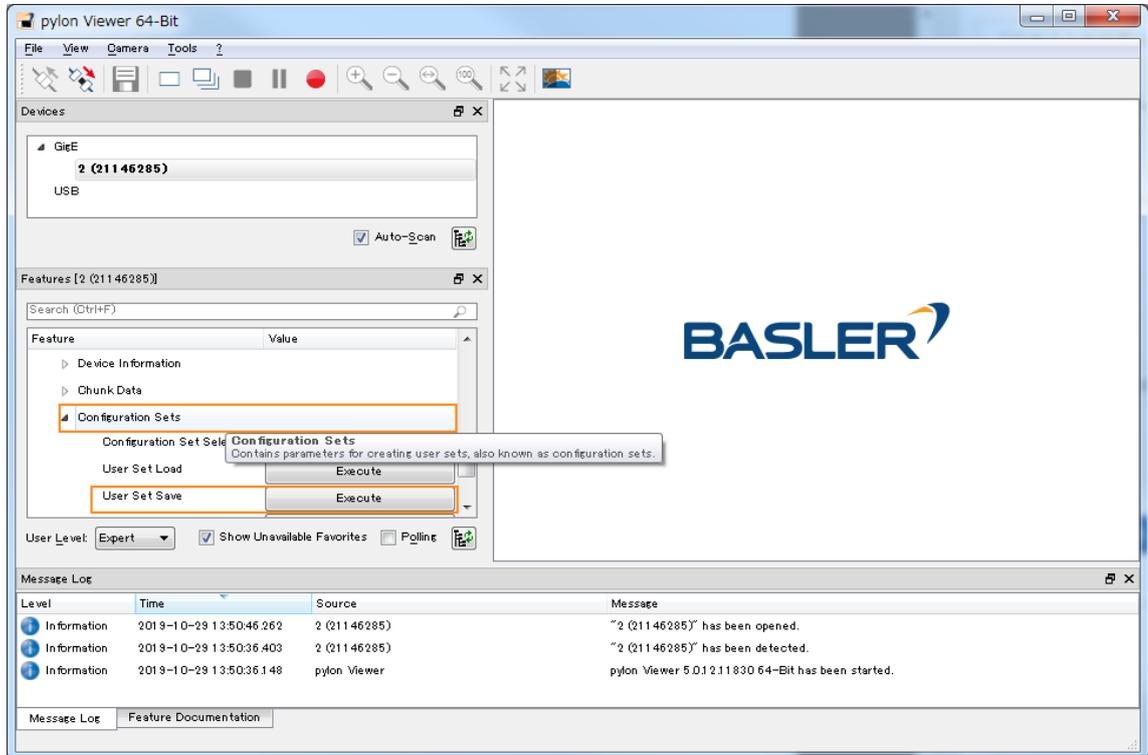


- “Feature” → “Acquisition Controls” を開きます。
“Trigger Mode”を “On”にし、使用するモードに応じて “Trigger Source”を選択します。



- その他の使用条件(シャッター、スピード、ゲインなど)を設定します。

- “Configuration Set Selector”で設定を保存するセットを選択します。 “User Set Save” の “Execute” を押して設定した条件を保存します。この操作を行わないとカメラの設定は保存されません。



カメラの電源を入れた時は、“Default Startup Set”で指定されたセット(上図では “User Set2”)の設定で立ち上がります。

付録B. Pylon6(SDK Version 6.1.0) セットアップ手順

Pylon SDK Version 6.1.x の最新バージョンである 6.1.0 のセットアップ手順を記述します。

1. SDK のインストール

- Basler 社より Windows 用ソフトウェアスイート Pylon 6.1.0 をダウンロードしてください。
- ダウンロードした”Basler_pylon_6.1.0.19674.exe”を実行してください。

ダウンロードおよびインストール手順は付録 A を参照してください。

2. GigE カメラの IP アドレスと ID の設定

- インストールされた pylon IP Configurator を起動し、設定したいカメラの “IP Address”, “ Subnet Mask”, “Gateway”, “Device User ID”を入力し, “Save”ボタンを押します。

Name	Device User ID	Serial Number	MAC Address	Status	IP Configuration	IP Address	Subnet Mask
イーサネット	2	21146285	00:30:53:11:7D:AD	OK	Static IP	192.168.0.88	255.255.255.0
Wi-Fi							
vEthernet (Default Switch)							

Static IP **2 (21146285)** Refresh

IP Address: Vendor: Basler
Subnet Mask: Model Name: acA1300-30gc
Gateway: Device User ID: 2
 DHCP Serial Number: 21146285
 Auto IP (LLA) MAC Address: 00:30:53:11:7D:AD
Device User ID: IP Configuration: Static IP
IP Address: 192.168.0.88
Subnet Mask: 255.255.255.0
Gateway: 0.0.0.0

Save

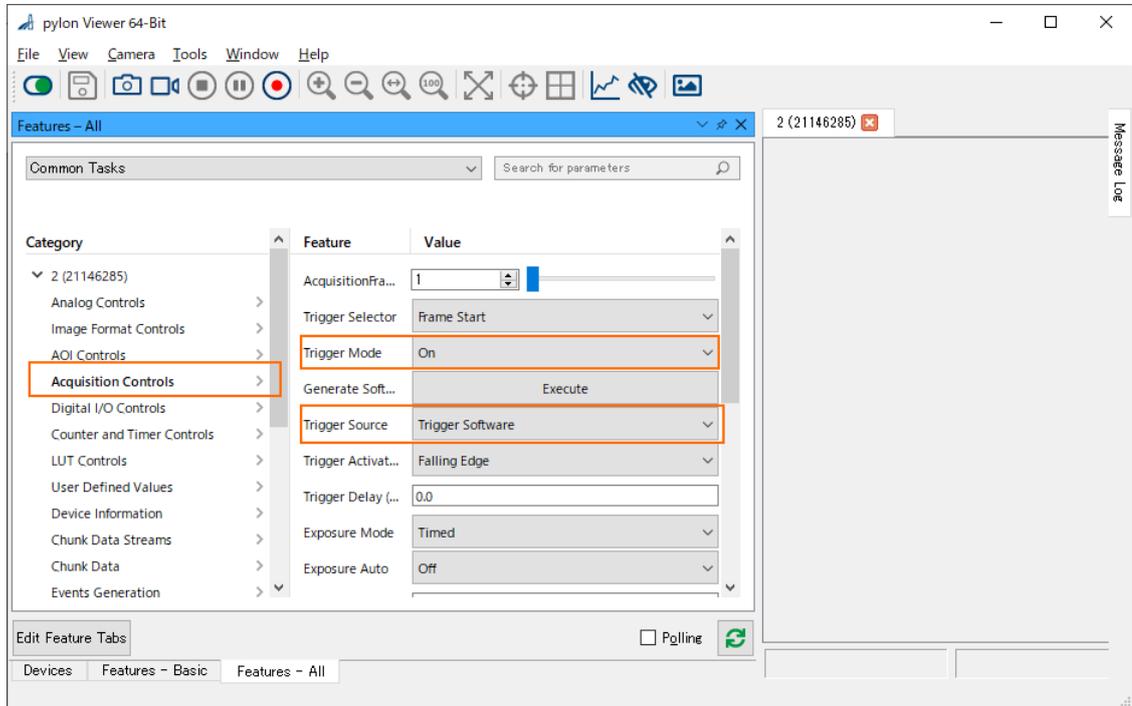
Device User ID の範囲は 0 - 4 です。

User ID 0 - 4 はコントローラのイメージ格納領域の ID201 - 205 に対応しています。

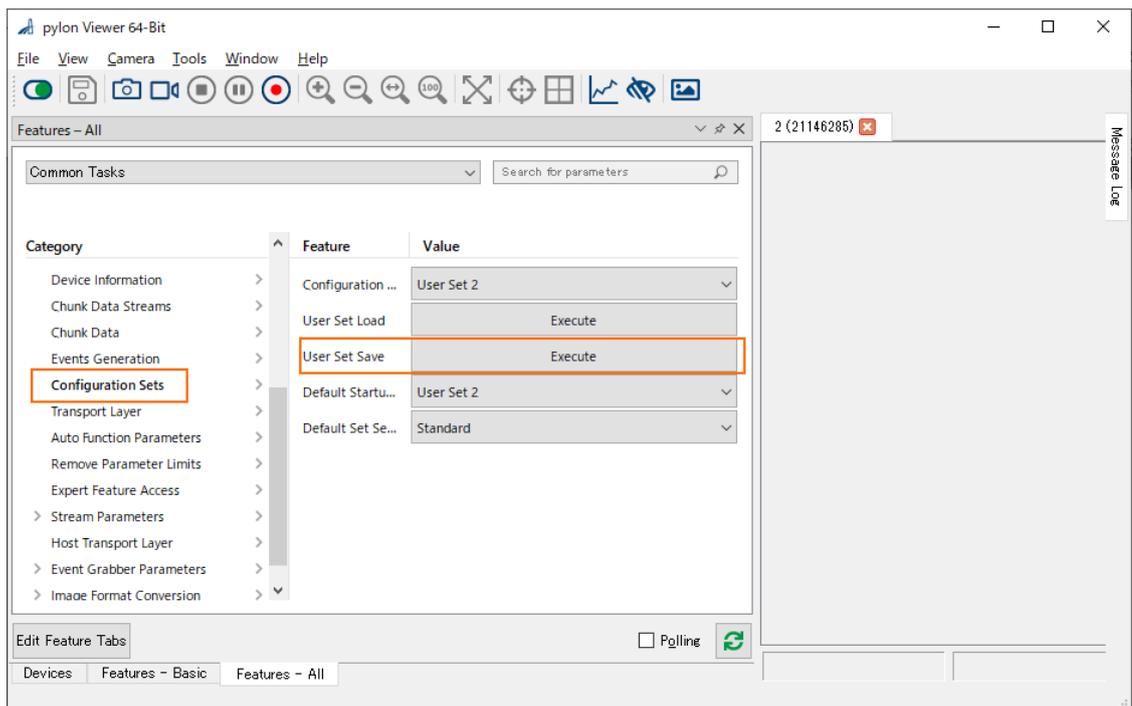
カメラを複数台使用するときは ID を重複しないようにしてください。

3. カメラの動作設定

- “Feature All” → “Acquisition Controls” を開きます。
“Trigger Mode”を “On”にし、使用するモードに応じて “Trigger Source”を選択します。



- その他の使用条件(シャッター, スピード, ゲインなど)を設定します。
- “Configuration Set Selector”で設定を保存するセットを選択します. “User Set Save” の “Execute” を押して設定した条件を保存します. この操作を行わないとカメラの設定は保存されません.



カメラの電源を入れた時は、“Default Startup Set”で指定されたセット(上図では “User Set2”)の設定で立ち上がります。

付録C. Pylon API 対応表

PylonGigE プロバイダは、Pylon API の CBaslerGigEInstantCamera クラスを使用して GigE カメラとの通信を行っています。

ここでは、CaoController::Execute()の各コマンドと CBaslerGigEInstantCamera クラスメンバとの対応を示します。

表 3-2 CaoController::Execute メソッド-CBaslerGigEInstantCamera クラス対応表

Execute コマンド	CBaslerGigEInstantCamera クラスメンバ
Start	StartGrabbing
Stop	StopGrabbing
GetImage	RetrieveResult (画像取込み中) GrabOne (画像取込み停止中)
GetLastErrorDescription	-
ExecSoftTrigger	ExecuteSoftwareTrigger
GetSensorWidth	SensorWidth
GetSensorHeight	SensorHeight
GetWidthMax	WidthMax
GetHeightMax	HeightMax
GetDeviceVendorName	DeviceVendorName
GetDeviceModelName	GetDeviceModelName
GetDeviceManufacturerInfo	DeviceManufacturerInfo
GetDeviceVersion	DeviceVersion
GetDeviceFirmwareVersion	DeviceFirmwareVersion
GetDeviceID	DeviceID
PutDeviceUserID	DeviceUserID
GetDeviceUserID	
DeviceReset	DeviceReset
PutDeviceScanType	DeviceScanType
GetDeviceScanType	
PutWidth	Width

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	