V-Works for XG プロバイダ
KEYENCE XG シリーズ用プロバイダ

Version 1.0.4

ユーザーズ ガイド

July 01, 2019

【備考】
PC で VWXG プロバイダを利用するには, KEYENCE 社から提供されている「XG-8000/7000 シリーズ用の「V-Works for XG ActiveX コントロール」をインストールする必要があります.

【改版履歴】

バージョン	日付	内容
1.0.0	2011-12-08 2012-07-17	初版. ドキュメントのバージョンルールを変更
1.0.1	2012-11-15	TriggerAndGetResult コマンドを追加
1.0.2	2014-02-12	ReadInspectSetting コマンドを追加 Timeout オプション追記
1.0.3	2016-01-13	非同期処理のコマンドの追加 GetCommandResult コマンドの追加 RecievePacket コマンドの追加 TriggerAndGetResult コマンド修正
1.0.4	2019-07-01	プロバイダ切断時の不具合を修正

目次

1. はじめに 5

2. プロバイダの概要 6

2.1. 概要 6
2.2. メソッド・プロパティ 6
2.2.1. CaoWorkspace::AddController メソッド 6
2.2.1.1. Conn オプション 7
2.2.2. CaoController::Execute メソッド 7
2.2.3. エラーコード 7
2.2.4. CaoController::OnMessage イベント 8

3. コマンドリファレンス 10

3.1. リモートデスクトップコマンド 11
3.1.1. CaoController::Execute("StartRemoteDesktop") コマンド 11
3.1.2. CaoController::Execute("StopRemoteDesktop") コマンド 11
3.1.3. CaoController::Execute("UpdateRemoteDesktop") コマンド 12
3.1.4. CaoController::Execute("CaptureRemoteDesktop") コマンド 12
3.1.5. CaoController::Execute("ClearRemoteDesktop") コマンド 12
3.2. 結果ロギングコマンド 13
3.2.1. CaoController::Execute ("StartResultLog") コマンド 13
3.2.2. CaoController::Execute ("StopResultLog") コマンド 13
3.2.3. CaoController::Execute ("SetResultLogUnit") コマンド 14
3.2.4. CaoController::Execute ("GetResultLogData") コマンド 14
3.3. 無手順コマンド 15
3.3.1. CaoController::Execute ("ExecuteCommand") コマンド 15
3.3.2. CaoController::Execute ("ExecuteCommandAsync") コマンド 15
3.3.3. CaoController::Execute ("ChangeMode") コマンド 16
3.3.4. CaoController::Execute ("ChangeModeAsync") コマンド 16
3.3.5. CaoController::Execute ("ReadMode") コマンド 17
3.3.6. CaoController::Execute ("Reset") コマンド 17
3.3.7. CaoController::Execute ("Restart") コマンド 18
3.3.8. CaoController::Execute ("RestartAsync") コマンド 18
3.3.9. CaoController::Execute ("Trigger") コマンド 18
3.3.10. CaoController::Execute ("EnableTrigger") コマンド 19

3.3.11. CaoController::Execute ("ReadTriggerEnable") コマンド 19
3.3.12. CaoController::Execute ("WriteVariable") コマンド 20
3.3.13. CaoController::Execute ("ReadVariable") コマンド 20
3.3.14. CaoController::Execute ("ChangeInspectSetting") コマンド 21
3.3.15. CaoController::Execute ("ChangeInspectSettingAsync") コマンド 21
3.3.16. CaoController::Execute ("ReadInspectSetting") コマンド 22
3.3.17. CaoController::Execute ("ClearError") コマンド 22
3.3.18. CaoController::Execute ("TriggerAndGetResult") コマンド 23
3.3.19. CaoController::Execute ("GetCommandResult") コマンド 24
3.3.20. CaoController::Execute ("ReceivePacket") コマンド 24

1. はじめに

本書は KEYENCE 社製ビジョンシステムである XG シリーズと連動動作する V-Works for XG ActiveX コントロールの CAO プロバイダである, V-Works for XG プロバイダ（以下 VWXG プロバイダ）のユーザーズガイドです.

VWXG プロバイダは Ethernet 接続された XG シリーズコントローラと通信経由で連携処理や処理結果の通知を行います. PC で VWXG プロバイダを利用するには, KEYENCE 社から提供されている XG-8000/7000 シリーズ用の「V-Works for XG ActiveX コントロール」をインストールする必要があります.

本書は, このVWXGプロバイダの機能と実装されているメソッドについて説明します.

2. プロバイダの概要

2.1. 概要

VWXG プロバイダは，コマンドの実行方法として CaoController::Execute による方法を提供しています．[1]
CaoController::Execute では，シリアルインタフェースを利用しコマンドの送受信を行います．

表 2-1 VWXG プロバイダ

ファイル名	CaoProvVWXG.dll
ProgID	CaoProv.KEYENCE.VWXG
レジストリ登録[2]	regsvr32 CaoProvVWXG.dll
レジストリ登録の抹消	regsvr32 /u CaoProvVWXG.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

VWXG プロバイダでは AddController 時に，通信用の接続パラメータを参照し，通信の接続を行います．このときオプションで通信形態を指定します．

書式　AddController(<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])

bstrCtrlName　: [in] コントローラ名　任意
bstrProvName　: [in] プロバイダ名　固定値　=" CaoProv.KEYENCE.VWXG"
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します．

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

オプション	意味
Conn =<接続パラメータ>	必須．通信形態とその接続パラメータを設定します． 詳細は 2.2.1.1 を参照してください．
Timeout=[<タイムアウト時間>]	送受信時のタイムアウト時間を指定します．（デフォルト：500）

[1] V-Works for XG がインストールされていない環境で AddController を実行すると，実行環境が不安定になります．
[2] ORiN SDK でインストールした場合は手動で登録／抹消する必要はありません．

2.2.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します. ここで角括弧("[ ]")内は省略可能を示します. また, 各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値を示します.

- Ethernet デバイス

  "eth:<IP Address> [:<Port No>]"

  <IP Address> : 接続する XG-7000 シリーズの IP アドレス.

  例:"192.168.0.10", "192.168.0.11"

  <Port No> : 接続ポート番号. 8502, 8503, . . . 任意指定可能

2.2.2. CaoController::Execute メソッド

イーサネットを介して無手順方式でコマンドの送受信を行います. 第 1 引数にコマンド名, 第 2 引数にコマンドのパラメータを指定します. 各コマンドの詳細は 3 章コマンドリファレンスを参照してください.

書式 Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

bstrCommandName: [in] コマンド名

vntParam : [in] パラメータ

2.2.3. エラーコード

Ececute メソッド実行した際の XG-7000 シリーズからの処理結果は HRESULT として返されます.

正常に処理された場合(OK) : S_OK (0)

正常に処理されなかった場合(ER) : 0x80100000 + 戻り値

例:ChangeMode を実行したとき.

hr = 0x801003EA : 引数の値が範囲外です.

エラー内容については XG-7000 シリーズ用 V-Works for XG ActiveX コントロールのリファレンスマニュアルを参照してください.

表 2-3 エラーコード一覧

エラー名	エラー番号	説明
E_BASED_VWXG	0x80100000 | VWXGエラー	VWXG シリーズ固有エラー
E_ENABLED_CANCEL	0x80101000	キャンセル処理に失敗しました
E_EXECUTING	0x80102000	コマンド実行中に別コマンドを実行しました

E_GET_COMMAND_RESULT	0x80103000	同期コマンド実行後に GetCommandResult コマンドを実行しました

2.2.4. CaoController::OnMessage イベント

XG-7000 シリーズのイベントが発生したときに，CAO の OnMessage イベントが発生します．

Message::Number プロパティには，イベント種別が格納されます．

Message::Value プロパティには，イベントの詳細データが格納されます．

XG-7000 シリーズの各イベントの詳細については，XG-7000 シリーズ用 V-Works for XG ActiveX コントロールのリファレンスマニュアルを参照してください．

表 2-4 イベント種別一覧

種別	説明 （XG-7000 シリーズイベント）	Value の内容
0x000	リモートデスクトップの画面更新 （OnRemoteDesktopUpdated）	(<iState>) < iState > = VT_I2：処理結果 0： 処理成功 1100： 通信例外
0x001	結果出力データの受信 （OnResultLogDataReceived）	(<iState>, <iDriveNo>, <iSettingNo>, <iUnitId>, <bstrDstFile>) < iState > = VT_I2：処理結果 0： 処理成功 1001： 処理失敗 1004： 残量なし 1100： 通信例外 < iDriveNo > = VT_I2：SD カード番号 1：SD1 2：SD2 -1：取得失敗 < iSettingNo > = VT_I2：現在の検査設定 No.（失敗時は-1） （0～999） < iUnitId > = VT_I2：結果出力したユニット ID（失敗時は-1） （0～999） < bstrDstFile > = VT_BSTR：格納先ファイルのフルパス名 （失敗時は空文字）
0x002	結果ロギングの停止	(<iState>)

	（OnResultLogStopped）	< iState > = VT_I2：処理結果 0： 処理成功 1100： 通信例外
0x00A	無手順コマンドの実行 （OnCommandFinished）	(<iState>, <iKind>) < iState > = VT_I2：処理結果 0： 処理成功 < iKind > = VT_I2：応答対象のコマンド種別

< iState >のエラーについては，XG-7000 シリーズ用 V-Works for XG ActiveX コントロールリファレンスマニュアルの各コマンドの「イベント処理結果」の項目を参照してください.

応答対象のコマンド一覧については，XG-7000 シリーズ用 V-Works for XG ActiveX コントロールリファレンスマニュアルを参照してください.

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. 各コマンドの詳細動作については KEYENCE 社の XG-7000 シリーズ用 V-Works for XG ActiveX コントロールのリファレンスマニュアルの API 一覧を参照してください.

表 3-1 CaoController::Execute コマンド一覧

XG-7000 シリーズ コマンド	コマンド	機能	
リモートデスクトップコマンド			
StartRemoteDesktop	StartRemoteDesktop	リモートデスクトップを開始します	P11
StopRemoteDesktop	StopRemoteDesktop	リモートデスクトップを停止します	P11
UpdateRemoteDesktop	UpdateRemoteDesktop	リモートデスクトップを更新します	P12
CaptureRemoteDesktop	CaptureRemoteDesktop	リモートデスクトップ画面のファイル保存をします	P12
ClearRemoteDesktop	ClearRemoteDesktop	リモートデスクトップ画面をクリアします	P12
結果ロギングコマンド			
StartResultLog	StartResultLog	結果ロギングを開始します	P13
StopResultLog	StopResultLog	結果ロギングを停止します	P13
SetResultLogUnit	SetResultLogUnit	結果出力データ取得対象ユニットを指定します	P14
GetResultLogData	GetResultLogData	結果出力データを取得します	P14
無手順コマンド			
ExecuteCommand	ExecuteCommand	無手順コマンドを実行します	P15
	ExecuteCommandAsync	無手順コマンドを実行します(非同期版)	P15
ChangeMode	ChangeMode	運転／停止モードを変更します	P16
	ChangeModeAsync	運転／停止モードを変更します(非同期版)	P16
ReadMode	ReadMode	運転／停止モードを読み出します	P17
Reset	Reset	リセットします	P17
Restart	Restart	フローの先頭へ復帰します	P18
	RestartAsync	フローの先頭へ復帰します(非同期版)	P18
Trigger	Trigger	トリガを発行します	P18
EnableTrigger	EnableTrigger	トリガ入力を許可／禁止します	P19
ReadTriggerEnable	ReadTriggerEnable	トリガ入力許可状態を読み出します	P19
WriteVariable	WriteVariable	変数値を書き込みます	P20
ReadVariable	ReadVariable	変数値を読み出します	P20
ChangeInspectSetting	ChangeInspectSetting	検査設定 No.を切り換えます	P21
	ChangeInspectSettingAsync	検査設定 No.を切り換えます(非同期版)	P21
ReadInspectSetting	ReadInspectSetting	検査設定 No.を取得します	P22
ClearError	ClearError	システムエラーをクリアします	P22

独自拡張コマンド			
―	TriggerAndGetResult	トリガを発行し，画像処理結果を取得します	P23
―	GetCommandResult	非同期コマンドの戻り値を取得します	P24
―	RecievePacket	パケットの受信を行います	P24

3.1. リモートデスクトップコマンド

3.1.1. CaoController::Execute("StartRemoteDesktop") コマンド

リモートデスクトップ機能を開始します.

自動更新モードを指定した場合，画面更新が終了するごとに OnRemoteDesktopUpDated イベントが発生します. 手動更新モードを指定した場合，画面転送されません. UpdateRemoteDesktop 関数を呼び出して，画面転送してください.

書式　StartRemoteDesktop(< lMode >)

< lMode > : リモートデスクトップ機能の動作モードを指定します(VT_I4)
0:自動更新モード
1:手動更新モード

戻り値 : なし

リモートデスクトップ機能を自動更新モードで開始する場合の例を以下に示します.

使用例

```
Dim lMode as long
lMode = 0

caoCtrl.Execute "StartRemoteDesktop", lMode
```

3.1.2. CaoController::Execute("StopRemoteDesktop") コマンド

リモートデスクトップ機能を停止します.

書式　StopRemoteDesktop( )

引数 : なし

戻り値 : なし

リモートデスクトップ機能を停止する場合の例を以下に示します.

使用例

```
caoCtrl.Execute "StopRemoteDesktop"
```

3.1.3. CaoController::Execute("UpdateRemoteDesktop") コマンド

コントローラのモニタ画面を取得し，リモートデスクトップ画面を更新します．

画面更新が終了すると，OnRemoteDesktopUpdated イベントが発生します．

書式　UpdateRemoteDesktop( )

引数　：　なし

戻り値　：　なし

リモートデスクトップ画面を更新する場合の例を以下に示します．

使用例

```
caoCtrl.Execute "UpdateRemoteDesktop"
```

3.1.4. CaoController::Execute("CaptureRemoteDesktop") コマンド

現在表示されているリモートデスクトップ画面を，ビットマップ形式の画像ファイルに格納します．

書式　CaptureRemoteDesktop(< bstrFile >)

bstrFile　：　出力先のビットマップ形式の画像ファイルのフルパスを指定します（VT_BSTR）

戻り値　：　なし

リモートデスクトップ画面を"D:¥Temp¥Img.bmp "ファイルで保存する場合の例を以下に示します．

使用例

```
Dim bstrFile as string
bstrFile = "D:¥Temp¥Img.bmp"

caoCtrl.Execute "CaptureRemoteDesktop", bstrFile
```

3.1.5. CaoController::Execute("ClearRemoteDesktop") コマンド

現在表示されているリモートデスクトップ画面を初期化（黒画面）します．

書式　ClearRemoteDesktop( )

引数　：　なし

戻り値　：　なし

リモートデスクトップ画面を初期化する場合の例を以下に示します．

使用例

```
caoCtrl.Execute "ClearRemoteDesktop"
```

3.2. 結果ロギングコマンド

3.2.1. CaoController::Execute ("StartResultLog") コマンド

結果ロギング機能を開始します.

結果出力データを受信すると, OnResultLogDataReceived イベントにより通知されます.

書式　StartResultLog(< iMode >, < bstrFolder >)

< iMode > : [in] 受信したデータをファイル格納するか否かを指定します. (VT_I2)
0 : 受信したデータをファイルに格納します.
1 : 受信したデータをファイルに格納しません.

< bstrFolder > : [in] 受信したデータファイルを格納するベースとなるフォルダ名をフルパスで指定します(最大 255 文字). iMode 引数が 0 の場合のみ参照します.
このフォルダ下に SD1, SD2 フォルダを作成し, 命名規則に従ってファイルを格納します. (VT_BSTR)

ベースフォルダに「C:¥work」を設定し, 結果ロギングを開始する場合の例を以下に示します.

使用例

```
Dim bstrFolder as string
bstrFolder = "C:¥work"

caoCtrl.Execute "StartResultLog", Array(0, bstrFolder)
```

3.2.2. CaoController::Execute ("StopResultLog") コマンド

結果ロギング機能を停止します.

要求受付のみをおこない, 結果ロギング動作が停止すると OnResultLogStopped イベントが発生します.

書式　StopResultLog( )

引数 : なし
戻り値 : なし

結果ロギングを停止する場合の例を以下に示します.

使用例

```
caoCtrl.Execute "StopResultLog"
```

3.2.3. CaoController::Execute ("SetResultLogUnit") コマンド

GetResultLogData コマンドの取得対象となる結果出力ユニットを指定して，最新データを取得用に固定します．

書式　SetResultLogUnit(< iUnitId >)

< iUnitId >　：　結果出力ユニット ID を指定します（0～999）（VT_I2）

戻り値　：　なし

最新データを取得する取得対象ユニットに 5 番を指定する場合の例を以下に示します．

使用例

```
Dim iUnitId as integer
iUnitId = 5

caoCtrl.Execute "SetResultLogUnit", iUnitId
```

3.2.4. CaoController::Execute ("GetResultLogData") コマンド

受信した結果出力データを取得します．結果出力ユニットで設定された 4 つのインデックスで指定し出力文字列および数値データを取得します．

書式　[vntRet =] GetResultLogData(< iItemNo >, < iArrayNo >, < iMemberNo >)

< iItemNo >　：　結果出力ユニット内の出力項目番号を指定します（0～255）（VT_I2）

< iArrayNo >　：　出力項目の配列番号を指定します（0 開始で先頭から何番目かを指定します）．出力項目に配列を設定していない場合は，0 を指定します（VT_I2）

< iMemberNo >　：　位置型，円型，直線型変数のメンバ番号を指定します（0～2）
スカラ型変数の場合は 0 を指定します（VT_I2）
0 ： 位置型:X，円型:CX，直線型:T
1 ： 位置型:Y，円型:CY，直線型:RH
2 ： 円型:CR

< vntRet >　：　[out] 結果出力ユニットで指定した書式の文字列と，出力された値を数値で格納します（VT_VARIANT）

出力項番の 0 番（スカラ型変数）の値を取得する場合の例を以下に示します．

使用例

```
Dim iItemNo as integer
Dim iArrayNo as integer
Dim iMemberNo as integer
Dim vntResult as variant
iItemNo = 0
iArrayNo = 0
iMemberNo = 0

vntResult = caoCtrl.Execute "GetResultLogData", Array(iItemNo, iArrayNo, iMemberNo)
```

3.3. 無手順コマンド

3.3.1. CaoController::Execute ("ExecuteCommand") コマンド

指定した無手順コマンドを実行します. コマンド実行の成否にかかわらずコマンドの応答を取得します.

サポートする無手順コマンドについては, XG-7000 シリーズ用 V-Works for XG ActiveX コントロールリファレンスマニュアルを参照してください.

書式　[<vntRet> =]ExecuteCommand(< bstrCommand >)

bstrCommand	:	コマンドの文字列を指定します(VT_BSTR)
vntRet	:	コマンドの応答を返します. 取得に失敗した場合は文字列が格納されません(VT_BSTR)

無手順コマンドを指定し, XG-7000 を運転モードに切替える場合の例を以下に示します.

使用例

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("ExecuteCommand", "RO")
```

3.3.2. CaoController::Execute ("ExecuteCommandAsync") コマンド

指定した無手順コマンドを非同期で実行します.

コマンド実行の成否や戻り値は, GetCommandResult コマンドで取得します.

サポートする無手順コマンドについては, XG-7000 シリーズ用 V-Works for XG ActiveX コントロールリファレンスマニュアルを参照してください.

コマンドの戻り値は GetCommandResult コマンドで取得し, 確認してください. GetCommandResult コマンドの詳細については, 3.3.19.CaoController::Execute ("GetCommandResult") コマンドを参照ください.

書式　ExecuteCommandAsync(< bstrCommand >)

bstrCommand	:	コマンドの文字列を指定します(VT_BSTR)
戻り値	:	なし

無手順コマンドを指定し, XG-7000 を運転モードに切替える場合の例を以下に示します.

使用例

```
Dim vntResult as variant

caoCtrl.Execute("ExecuteCommandAsync", "R0")

‘ ExecuteCommandAsync コマンドの戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")

‘ vntResult : 無手順コマンドの戻り値 (BSTR)
```

3.3.3. CaoController::Execute ("ChangeMode") コマンド

運転モードまたは, 停止モードに移行します.

書式　ChangeMode(< iMode >)

iMode	:	[in] 変更先のモードを指定します 0 : 停止モード 1 : 運転モード
戻り値	:	なし

XG-7000 を運転モードに切替える場合の例を以下に示します.

使用例

```
caoCtrl.Execute "ChangeMode", 1
```

3.3.4. CaoController::Execute ("ChangeModeAsync") コマンド

非同期で, 運転モードまたは, 停止モードに移行します.

コマンドの戻り値は GetCommandResult コマンドで取得し, 確認してください. GetCommandResult コマンドの詳細については, 3.3.19.CaoController::Execute ("GetCommandResult") コマンドを参照ください.

書式　ChangeModeAsync(< iMode >)

iMode	:	[in] 変更先のモードを指定します 0 : 停止モード 1 : 運転モード
戻り値	:	なし

XG-7000 を運転モードに切替える場合の例を以下に示します.

使用例

```
Dim vntResult as variant

caoCtrl.Execute "ChangeModeAsync", 1
vntResult = caoCtrl.Execute("GetCommandResult")

‘ vntResult : Empty
```

3.3.5. CaoController::Execute ("ReadMode") コマンド

現在の動作モード(運転モード, 停止モード, リモートキャプチャモード)を取得します.

書式　[< vntRet > =] ReadMode( )

引数　：　なし

< vntRet >　：　[out] 現在の動作モードが格納されます. 取得に失敗した場合は-1が格納されます. (VT_I2)

0 ： 停止モード

1 ： 運転モード

2 ： リモートキャプチャモード

XG-7000(運転モード)のモードを取得する場合の例を以下に示します.

使用例

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("ReadMode")

‘ vntResult : 1
```

3.3.6. CaoController::Execute ("Reset") コマンド

コントローラをリセットします.

書式　Reset( )

引数　：　なし

戻り値　：　なし

コントローラをリセットする場合の例を以下に示します.

使用例

```
caoCtrl.Execute "Reset"
```

3.3.7. CaoController::Execute ("Restart") コマンド

スタートユニットの次のユニットにジャンプします.

書式　Restart( )

引数　：　なし

戻り値　：　なし

Restart コマンドを使用する例を以下に示します.

使用例

```
caoCtrl.Execute "Restart"
```

3.3.8. CaoController::Execute ("RestartAsync") コマンド

非同期でスタートユニットの次のユニットにジャンプします.

コマンドの戻り値は GetCommandResult コマンドで取得し, 確認してください. GetCommandResult コマンドの詳細については, 3.3.19.CaoController::Execute ("GetCommandResult") コマンドを参照ください.

書式　RestartAsync( )

引数　：　なし

戻り値　：　なし

RestartAsync コマンドを使用する例を以下に示します.

使用例

```
Dim vntResult as variant

caoCtrl.Execute "RestartAsync"

' RestartAsync コマンドの戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : 戻り値 (Empty)
```

3.3.9. CaoController::Execute ("Trigger") コマンド

トリガを発行します.

書式　Trigger(< iTriggerNo >)

iTriggerNo　：　発行対象のトリガ番号を指定します.（VT_I2）

1～4 ： トリガ1～4

-1 ： 全トリガ

戻り値 : なし

トリガ番号1を発行する場合の例を以下に示します.

使用例

```
Dim iTriggerNo as integer
iTriggerNo = 1

caoCtrl.Execute "Trigger", iTriggerNo
```

3.3.10. CaoController::Execute ("EnableTrigger") コマンド

トリガ入力の許可／禁止を設定します.

書式 EnableTrigger(< iMode >)

iMode : [in] トリガの許可／禁止を設定します.（VT_I2）
0 : トリガ禁止
1 : トリガ許可

戻り値 : なし

トリガ入力を禁止に設定する場合の例を以下に示します.

使用例

```
Dim iMode as integer
iMode = 0

caoCtrl.Execute "EnableTrigger", iMode
```

3.3.11. CaoController::Execute ("ReadTriggerEnable") コマンド

現在のトリガ許可／禁止を取得します.

書式 [< vntRet > =] ReadTriggerEnable( )

引数 : なし

vntRet : トリガの許可, 禁止状態が格納されます. 取得に失敗した場合は-1が格納されます.（VT_I2）
0 : トリガ禁止状態
1 : トリガ許可状態

XG-7000(トリガ許可状態)のトリガ状態を取得する場合の例を以下に示します.

使用例

```
Dim vntResult as variant

‘ vntResult = caoCtrl.Execute "ReadTriggerEnable"
```

3.3.12. CaoController::Execute ("WriteVariable") コマンド

指定したスカラ型変数(グローバル変数, ローカル変数)へ値を書き込みます.

書式　WriteVariable(< bstrName >, < dblValue >, < iSyncMode >)

< bstrName >	:	スカラ型の変数名を半角文字で指定します.(VT_BSTR)
< dblValue >	:	変数に書き込む数値を指定します.(VT_R8)
< iSyncMode >	:	フローと同期して反映するかを指定します.(VT_I2)
戻り値	:	なし

スカラ型変数「#MyVar」へ値の書き込みをする場合の例を以下に示します.

使用例

```
Dim bstrName as string
Dim dblValue as double
bstrName = "#MyVar"
dblValue = 100

caoCtrl.Execute "WriteVariable", Array(bstrName, dblValue, 0)
```

3.3.13. CaoController::Execute ("ReadVariable") コマンド

指定されたスカラ型変数の値を取得します.

書式　[< vntRet > =] ReadVariable(< bstrName >)

bstrName	:	スカラ型の変数名を半角文字で指定します.(VT_BSTR)
vntRet	:	変数値が格納されます. 取得に失敗した場合は-1.0 が格納されます.(VT_R8)

スカラ型変数「#MyVal(=100)」の値を取得する場合の例を以下に示します.

使用例

```
Dim bstrName as string
Dim dblValue as double
bstrName = "#MyVar"

dblValue = caoCtrl.Execute "ReadVariable", bstrName
```

3.3.14. CaoController::Execute ("ChangeInspectSetting") コマンド

指定された SD カードの検査設定 No.に設定を切り換えます.

書式　[< vntRet > =] ChangeInspectSetting (< iDriveNo >, < iSettingNo >)

iDriveNo	:	SD カード番号を指定します.（VT_I4）
		1 ： SD1
		2 ： SD2
iSettingNo	:	検査設定 No.を指定します.（0～999）(VT_I4)
戻り値	:	なし

SD カード(SD1)の検査設定 No.1 に設定を切り換える場合の例を以下に示します.

使用例

```
Dim iDriveNo as integer
Dim iSettingNo as integer
iDriveNo = 1
iSettingNo = 1

Call caoCtrl.Execute("ChangeInspectSetting", Array(iDriveNo, iSettingNo))
```

3.3.15. CaoController::Execute ("ChangeInspectSettingAsync") コマンド

非同期で指定された SD カードの検査設定 No.に設定を切り換えます.

コマンドの戻り値は GetCommandResult コマンドで取得し，確認してください．GetCommandResult コマンドの詳細については，3.3.19.CaoController::Execute ("GetCommandResult") コマンドを参照ください.

書式　[< vntRet > =] ChangeInspectSettingAsync (< iDriveNo >, < iSettingNo >)

iDriveNo	:	SD カード番号を指定します.（VT_I4）
		1 ： SD1
		2 ： SD2
iSettingNo	:	検査設定 No.を指定します.（0～999）(VT_I4)
戻り値	:	なし

SD カード(SD1)の検査設定 No.1 に設定を切り換える場合の例を以下に示します.

使用例

```
Dim vntResult as variant
Dim iDriveNo as integer
Dim iSettingNo as integer
iDriveNo = 1
iSettingNo = 1

Call caoCtrl.Execute("ChangeInspectSettingAsync", Array(iDriveNo, iSettingNo))
```

```
' ChangeInspectionSettingAsync コマンドの戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : 戻り値 (Empty)
```

3.3.16. CaoController::Execute ("ReadInspectSetting") コマンド

現在使用している検査設定 No.を取得します.

書式 [< vntRet > =] ReadInspectSetting ()

引数	:	[out] なし
<vntRet>	:	[out] 設定値(<iDriveNo>, <iSettingNo>) (VT_I4 | VT_ARRAY)
		<iDriveNo>: SD カード番号
		<iSettingNo>: 検査設定 No.(0～999)

使用例を以下に示します.

使用例

```
Dim vntRet as Variant
vntRet = caoCtrl.Execute("ChangeInspectSetting")

' vntRet(0) : SD カード No
' vntRet(1) : 検査設定 No
```

3.3.17. CaoController::Execute ("ClearError") コマンド

指定した種別のエラー状態とエラーコードをクリアします.

書式 ClearError(< iErrorNo >)

iErrorNo	:	エラー状態種別を指定します.(VT_I2)
		0 : %Error0, %Error0Code をクリアします
		1 : %Error1, %Error1Code をクリアします
戻り値	:	なし

エラー状態種別が 0 のエラーをクリアする場合の例を以下に示します.

使用例

```
Dim iErrorNo as integer
iErrorNo = 0

caoCtrl.Execute "ClearError", iErrorNo
```

3.3.18. CaoController::Execute ("TriggerAndGetResult") コマンド

指定したトリガを発行し，画像処理結果を取得します．

書式　TriggerAndGetResult(< iTriggerNo >)

iTriggerNo　：　発行対象のトリガ番号を指定します．(VT_I2)
1～4 ： トリガ1～4
-1 ： 全トリガ

戻り値　：　[out] 結果出力ユニットで設定した出力データを格納します (VT_BSTR)

全トリガを発行し，結果を取得する場合の例を以下に示します．

使用例

```
Dim iTrigger as integer
Dim vntRet as Variant

iTrigger = -1

vntRet = caoCtrl.Execute("TriggerAndGetResult", iTrigger)
```

3.3.19. CaoController::Execute ("GetCommandResult") コマンド

非同期コマンドの完了待ちを行い，非同期コマンドの戻り値を取得します．

戻り値がない非同期コマンドを実行した場合，戻り値はありません．

また，同期コマンドの後で使用した場合は，GetCommandResult コマンド実行時に結果取得エラー(0x80103000)になり，戻り値はありません．

非同期コマンドの実行でエラーが発生した場合，非同期コマンドの実行時にはエラーは発生せず，

GetCommandResult コマンド実行時にエラーとなります．

非同期コマンドの完了待ちの際，設定されているタイムアウト時間以内に応答がない場合，

タイムアウトエラー（0x80000900）が発生します．

非同期コマンド実行後に別のコマンドを実行した場合は，先に実行した非同期コマンドの結果は削除されますのでご注意ください．

書式　　[<vntRet> =]GetCommandResult ()

引数　　：　なし

vntRet　　：　[out] 非同期コマンドの戻り値（VT_VARIANT）

非同期で検査を実行した場合の，戻り値を取得する例を以下に示します．

使用例

```
Dim vntResult as variant
caoCtrl.Execute "ReStartAsync"
vntResult = caoCtrl.Execute("GetCommandResult")
```

3.3.20. CaoController::Execute ("ReceivePacket") コマンド

パケットの受信を行います．

受信バッファにパケットがある場合は，受信バッファからパケットの取得を行います．

書式　　<vntRet> = RecievePacket ()

引数　　：　なし

vntRet　　：　[out] 受信パケット(VT_BSTR)

Trigger コマンドとの組み合わせ例を下記に示します．

使用例

```
Dim strRet as String

Call caoCtrl.Execute("Trigger", 1)
strRet = caoCtrl.Execute("RecievePacket")
```