

CVX プロバイダ KEYENCE CV-X シリーズ用プロバイダ

Version 1.0.3

ユーザーズ ガイド

August 29, 2019

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2014-02-11	初版.
1.0.1	2016-01-20	非同期処理の追加
1.0.2	2017-07-05	CV-X200 コントローラ対応コマンドの追加 <ul style="list-style-type: none"> •ResetTrigger •ReadMode •SetShutterSpeed •SetSensitivity •SetTriggerDelay •EnterImg •UpdateCapturePos
1.0.3	2019-07-09	計測コマンドの追加 <ul style="list-style-type: none"> •RegisterOneCharToLibrary •DeleteOneCharFromLibrary 計測値補正コマンドの追加 <ul style="list-style-type: none"> •ChangePreCorrectionMeasuredValue •WriteMeasuredValueCorrection •ReadMeasuredValueCorrection
1.0.3	2019-08-29	非同期コマンド使用時の注意点追加

【対応機器】

機種	バージョン	注意事項
CVX-100		
CVX-200		
CVX-400		
CVX-480		

目次

1. はじめに.....	5
1.1. 機器の設定.....	5
1.1.1. RS-232C 設定.....	6
1.1.2. Ethernet 設定.....	7
2. プロバイダの概要.....	8
2.1. 概要.....	8
2.2. メソッド・プロパティ.....	8
2.2.1. CaoWorkspace::AddController メソッド.....	8
2.2.1.1. Conn オプション.....	9
2.2.2. CaoController::Execute メソッド.....	10
2.2.3. エラーコード.....	10
3. コマンドリファレンス.....	11
3.1. トリガー.....	12
3.1.1. CaoController::Execute (“Trigger”) コマンド.....	12
3.2. コントローラ制御.....	13
3.2.1. CaoController::Execute (“ChangeMode”) コマンド.....	13
3.2.2. CaoController::Execute (“ChangeModeAsync”) コマンド (非推奨).....	13
3.2.3. CaoController::Execute (“Reset”) コマンド.....	14
3.2.4. CaoController::Execute (“Reboot”) コマンド.....	14
3.2.5. CaoController::Execute (“StoreSetting”) コマンド.....	14
3.2.6. CaoController::Execute (“ClearError”) コマンド.....	15
3.2.7. CaoController::Execute (“ChangeDisplayPattern”) コマンド.....	15
3.2.8. CaoController::Execute (“EnableTrigger”) コマンド.....	15
3.2.9. CaoController::Execute (“ResetTrigger”) コマンド.....	16
3.2.10. CaoController::Execute (“ReadMode”) コマンド.....	16
3.3. 検査設定切換.....	16
3.3.1. CaoController::Execute (“ChangeInspectSetting”) コマンド.....	16
3.3.2. CaoController::Execute (“ChangeInspectSettingAsync”) コマンド (非推奨).....	17
3.3.3. CaoController::Execute (“ReadInspectSetting”) コマンド.....	17
3.4. 撮像制御.....	18
3.4.1. CaoController::Execute (“ChangeLightVolume”) コマンド.....	18
3.4.2. CaoController::Execute (“SetShutterSpeed”) コマンド.....	18

3.4.3. CaoController::Execute ("SetSensitivity") コマンド	19
3.4.4. CaoController::Execute ("SetTriggerDelay") コマンド	19
3.5. 計測	20
3.5.1. CaoController::Execute ("ChangeExecuteCondition") コマンド	20
3.5.2. CaoController::Execute ("ReadExecuteCondition") コマンド	20
3.5.3. CaoController::Execute ("WriteCharReg") コマンド	20
3.5.4. CaoController::Execute ("ReadCharReg") コマンド	21
3.5.5. CaoController::Execute ("ChangeToolParameter") コマンド	21
3.5.6. CaoController::Execute ("ReadToolParameter") コマンド	22
3.5.7. CaoController::Execute ("ChangeFlawLevel") コマンド	22
3.5.8. CaoController::Execute ("ReadFlawLevel") コマンド	22
3.5.9. CaoController::Execute ("EnterImg") コマンド	23
3.5.10. CaoController::Execute ("UpdateCapturePos") コマンド	23
3.5.11. CaoController::Execute ("RegisterOneCharToLibrary") コマンド	24
3.5.12. CaoController::Execute ("DeleteOneCharFromLibrary") コマンド	24
3.6. 計測値補正	25
3.6.1. CaoController::Execute ("ChangePreCorrectionMeasuredValue") コマンド	25
3.6.2. CaoController::Execute ("WriteMeasuredValueCorrection") コマンド	25
3.6.3. CaoController::Execute ("ReadMeasuredValueCorrection") コマンド	27
3.7. 独自コマンド	28
3.7.1. CaoController::Execute ("ExecuteCommand") コマンド	28
3.7.2. CaoController::Execute ("ExecuteCommandAsync") コマンド (非推奨)	28
3.7.3. CaoController::Execute ("TriggerAndGetResult") コマンド	29
3.7.4. CaoController::Execute ("RecievePacket") コマンド	29
3.7.5. CaoController::Execute ("ClearPacket") コマンド	29
3.7.6. CaoController::Execute ("SetTimeout") コマンド	30
3.7.7. CaoController::Execute ("GetTimeout") コマンド	30
3.7.8. CaoController::Execute ("GetCommandResult") コマンド	30

1. はじめに

本書は KEYENCE 社製ビジョンシステムである CV-X シリーズ用の CAO プロバイダである, CVX プロバイダのユーザーズガイドです. CVX プロバイダは RS-232C または Ethernet で接続された CV-X シリーズコントローラと無手順方式でコマンドの実行や処理結果の通知を行います.

本書は, このCVX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.

1.1. 機器の設定

環境設定から使用する通信手段の設定を行ってください. 本プロバイダでは RS-232C かネットワークが使用できます. 外部入出力設定のメニューより, 接続する手段を選択して設定してください.

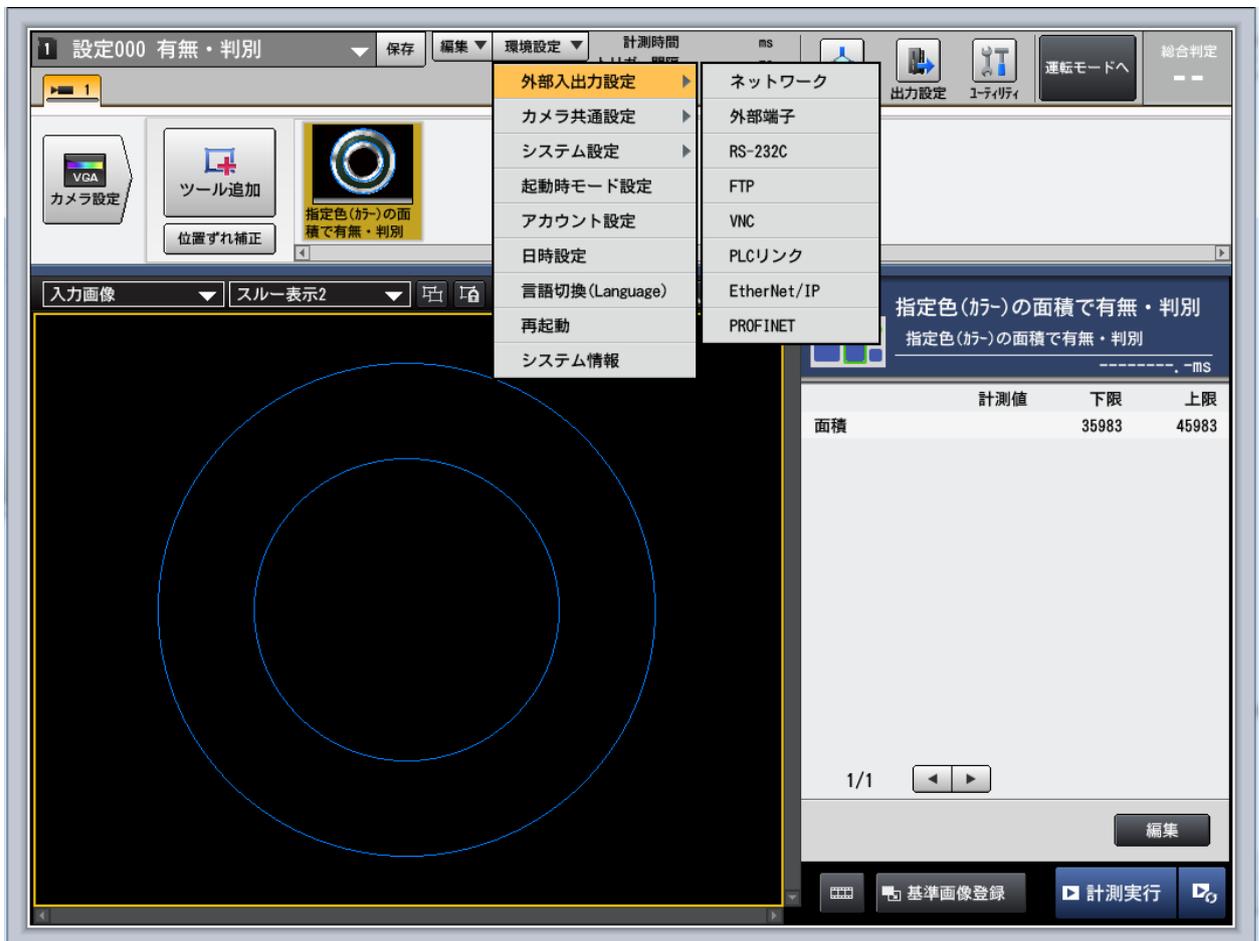


図 1-1 環境設定

1.1.1. RS-232C 設定

RS-232C の通信設定では、デリミター以外が設定変更可能になっています。デフォルト設定から変更した場合は、AddController の接続パラメータを変更してください(2.2.1.1 参照)。デリミターは"CR"を設定してください。



図 1-2 RS-232C 設定

1.1.2. Ethernet 設定

Ethernet の通信設定では、デリミター以外が設定変更可能になっています。デフォルト設定から変更した場合は、AddController の接続パラメータを変更してください(2.2.1.1 参照)。デリミターは"CR"を設定してください。

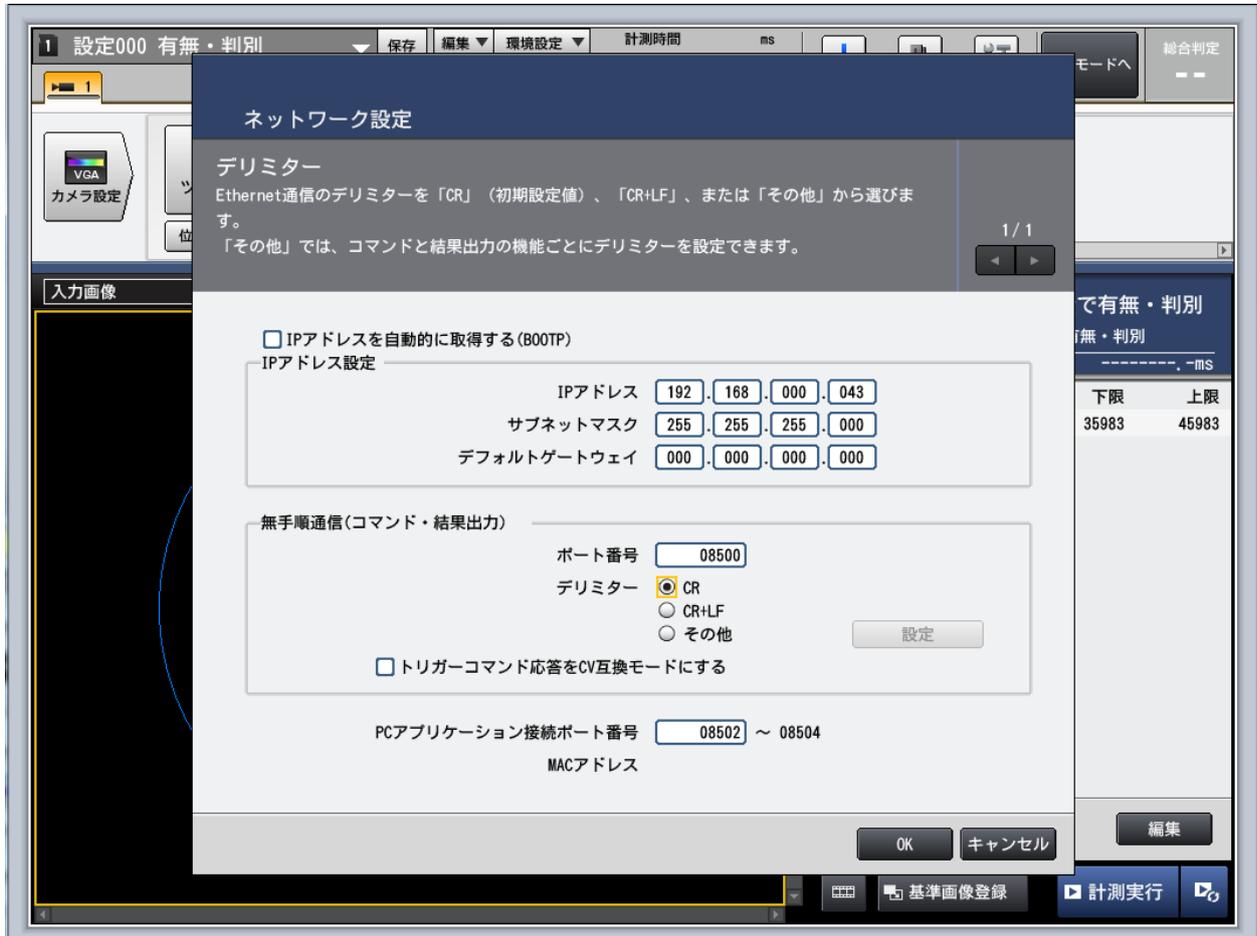


図 1-3 ネットワーク設定

2. プロバイダの概要

2.1. 概要

CVX 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。
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 CVX プロバイダ

ファイル名	CaoProvCVX.dll
ProgID	CaoProv.KEYENCE.CVX
レジストリ登録 ¹	regsvr32 CaoProvCVX.dll
レジストリ登録の抹消	regsvr32 /u CaoProvCVX.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

CVX 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.CVX”
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(ミリ秒)を指定します. (デフォルト: 500)

¹ 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 CVX 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
8500, 8501, . . . 任意指定可能

使用例

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng = New CaoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("CVX", "caoProv.KEYENCE.CVX", "",
"conn=eth:192.168.0.1, timeout=800")
```

- RS232C デバイス

"Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"

- <COM Port> : COM ポート番号
 '1'-COM1, '2'-COM2, ...
- <BaudRate> : 通信速度
 4800, 9600, 19200, 38400, 57600, 115200.
- <Parity> : パリティ
'N'-NONE, 'E'-EVEN, 'O'-ODD.
- <DataBits> : データビット数
 '7'-7bit, '8'-8bit.
- <StopBits> : ストップビット数
'1'-1bit, '2'-2bit.
- <Flow> : フロー制御
'0'-フロー制御なし, '1'-Xon/Xoff, '2'-ハードウェア制御.
 OR をとって指定できます。

使用例

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng = New CaoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("CVX", "caoProv.KEYENCE.CVX", "",
"conn=com:1")
```

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 章はじめに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

メソッドを実行した際の CVX シリーズからの処理結果は HRESULT として返されます。CVX シリーズ固有のエラーは 0x80108000 でマスクされて返されます。エラー内容については CVX シリーズのユーザーズマニュアルを参照してください。

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) : 0x80108000 + 戻り値

例: ChangeMode を実行したとき。

hr = 0x80108016 : 不正なパラメータが付与されています。

その他プロバイダ共通のエラー内容については ORIN2SDK プログラマーズユーザーズガイド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_CVERROR_CVERR	0x80108000 CVXエラー	CVX シリーズ固有エラー(2.2.3 参照)
E_CVERROR_LENGTH	0x80100000	パケット長エラー
E_CVERROR_PACKET	0x80100001	パケット異常エラー
E_COMMAND_EXECUTING	0x80100002	コマンド実行中に別コマンドを実行しました
E_GET_COMMAND_RESULT	0x80100003	同期コマンド実行後に GetCommandResult コマンドを実行しました

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、パラメータ範囲および内容については KEYENCE 社の CV-X シリーズユーザーズマニュアルの制御用通信コマンドリファレンスを参照してください。

表 3-1 CaoController::Execute コマンド一覧

CVX シリーズ 無手順コマンド	コマンド	機能	
トリガー			
T1, T2, T3, T4, TA	Trigger	トリガを発行します	P. 12
コントローラ制御			
R0, S0	ChangeMode	運転／停止モードを変更します	P. 13
	ChangeModeAsync	運転／停止モードを非同期で変更します (非推奨)	P. 13
RS	Reset	各種リセットを行います	P. 14
RB	Reboot	再起動を行います	P. 14
SS	StoreSetting	設定の保存を行います	P. 14
CE	ClearError	エラークリアを行います	P. 15
VW	ChangeDisplayPattern	運転画面の切り替えを行います	P. 15
TE	EnableTrigger	トリガ入力許可を変更します	P. 15
RE	ResetTrigger	計測を破棄して計測実行前の状態に戻ります	P. 16
RM	ReadMode	コントローラの状態を読み出します	P. 16
検査設定切替			
PW	ChangeInspectSetting	指定した設定 No に切り替えます	P. 16
	ChangeInspectSettingAsync	非同期で指定した設定 No に切り替えます (非推奨)	P. 17
PR	ReadInspectSetting	使用している設定 No を取得します	P. 17
撮像制御			
CLV	ChangeLightVolume	照明ボリューム値の設定を行います	P. 18
CSH	SetShutterSpeed	シャッター速度を変更します	P. 18
CSE	SetSensitivity	感度を変更します	P. 19
CTD	SetTriggerDelay	ディレイ時間を設定します	P. 19
計測			
EXW	ChangeExecuteCondition	実行条件の書き込みを行います	P. 20
EXR	ReadExecuteCondition	実行条件の読み出しを行います	P. 20
CW	WriteCharReg	判定文字列の書き換えを行います	P. 20
CR	ReadCharReg	判定文字列の読み出しを行います	P. 21

DW	ChangeToolParameter	判定条件の書き換えを行います	P. 21
DR	ReadToolParameter	判定条件の読み出しを行います	P. 22
SLW	ChangeFlawLevel	傷レベルの書き換えを行います	P. 22
SLR	ReadFlawLevel	傷レベルの読み出しを行います	P. 22
BS	EnterImg	基準画像を変更し、基準値を計算します	P. 23
CPW	UpdateCapturePos	撮像位置座標を更新します	P. 23
CA	RegisterOneCharToLibrary	読み取った文字を辞書に登録します	P. 24
CD	DeleteOneCharFromLibrary	指定した文字種の文字を辞書から削除します	P. 24
計測値補正			
MCC	ChangePreCorrectionMeasuredValue	任意の値の補正前計測値を計算します	P. 25
MCW	WriteMeasuredValueCorrection	補正值設定を書き換えて、補正值を計算します	P. 25
MCR	ReadMeasuredValueCorrection	設定している計測値補正の値を取得します	P. 27
独自拡張コマンド			
-	ExecuteCommand	無手順コマンドを実行します	P. 28
-	ExecuteCommandAsync	非同期で無手順コマンドを実行します (非推奨)	P. 28
-	TriggerAndGetResult	トリガを発行し、出力結果を受信します	P. 29
-	RecievePacket	パケットの受信を行います	P. 29
-	ClearPacket	バッファの受信パケットを破棄します	P. 29
-	SetTimeout	タイムアウト時間の設定を行います	P. 30
-	GetTimeout	タイムアウト時間の取得を行います	P. 30
-	GetCommandResult	非同期コマンドの戻り値を取得します	P. 30

3.1. トリガー

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

トリガを発行します。結果を出力設定している場合は RecievePacket コマンド等で受信してください。

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

```
caoCtrl.Execute "Trigger", 1
```

3.2. コントローラ制御

3.2.1. 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 (VT_UI4)

0 : 停止モード

1 : 運転モード

戻り値 : なし

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
```

3.2.2. CaoController::Execute (“ChangeModeAsync”) コマンド (非推奨)

運転モードまたは、停止モードに非同期で移行します。

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、3.7.8.CaoController::Execute (“GetCommandResult”) コマンドを参照ください。

コマンド使用時は、必ず GetCommandResult コマンドを実行後にプロバイダを切断してください。

非同期処理で異常が発生する場合があります。

書式

ChangeModeAsync(< iMode >)

iMode : [in] 変更先のモードを指定します (VT_UI4)

0 : 停止モード

1 : 運転モード

戻り値 : なし

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
‘ ChangeModeAsync コマンドの戻り値取得
vntResult = caoCtrl.Execute (“GetCommandResult”)
‘ vntResult : 戻り値 (Empty)
```

3.2.3. CaoController::Execute (“Reset”) コマンド

各種リセットを行います。

書式 Reset

 引数 : なし

 戻り値 : なし

例を以下に示します。

使用例

```
caoCtrl.Execute “Reset”
```

3.2.4. CaoController::Execute (“Reboot”) コマンド

現在の検査設定を保存し、再起動します。

書式 Reboot

 引数 : なし

 戻り値 : なし

例を以下に示します。

使用例

```
caoCtrl.Execute “Reboot”
```

3.2.5. CaoController::Execute (“StoreSetting”) コマンド

現在の検査設定、環境設定を保存します。

書式 StoreSetting

 引数 : なし

 戻り値 : なし

例を以下に示します。

使用例

```
caoCtrl.Execute "StoreSetting"
```

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

エラー状態をクリアします。エラー状態でないときも、正常終了します。



ClearError

引数 : なし

戻り値 : なし

例を以下に示します。



```
caoCtrl.Execute "ClearError"
```

3.2.7. CaoController::Execute ("ChangeDisplayPattern") コマンド

指定した運転画面に表示を切り換えます。



ChangeDisplayPattern(<IDisplay>, <INum>)

IDisplay : 指定画面種別 (VT_I4)

INum : 画面番号 (VT_I4)

戻り値 : なし

カメラ画面をカメラ 1 に切り換える例を以下に示します。



```
caoCtrl.Execute "ChangeDisplayPattern", Array(0, 1)
```

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

トリガ入力の許可／禁止を設定します。



EnableTrigger(< iMode >)

iMode : [in] トリガの許可／禁止を設定します。(VT_I2)

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

```
caoCtrl.Execute "EnableTrigger", 0
```

3.2.9. CaoController::Execute ("ResetTrigger") コマンド

実行中の計測撮像画像および計測結果を破棄して、計測実行前の状態に戻ります。

書式

ResetTrigger

引数 : [in]なし

戻り値 : [out]なし

例を以下に示します。

使用例

```
caoCtrl.Execute "ResetTrigger"
```

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

コントローラの状態(運転モード/設定モード)を読み出します。

書式

ReadMode

引数 : [in]なし

戻り値 : [out]コントローラの状態(VT_I4)

例を以下に示します。

使用例

```
caoCtrl.Execute "ReadMode"
```

3.3. 検査設定切換**3.3.1. 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

ChangeInspectSetting (< iDriveNo >, < iSettingNo >)

iDriveNo : [in] SD カード番号を指定します。(VT_I4)
 iSettingNo : [in] 検査設定 No.を指定します。(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

```
call caoCtrl.Execute("ChangeInspectSetting", Array(1, 1))
```

3.3.2. CaoController::Execute ("ChangeInspectSettingAsync") コマンド (非推奨)

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、3.7.8.CaoController::Execute ("GetCommandResult") コマンドを参照ください。

コマンド使用時は、必ず GetCommandResult コマンドを実行後にプロバイダを切断してください。

非同期処理で異常が発生する場合があります。

書式

ChangeInspectSettingAsync (< iDriveNo >, < iSettingNo >)

iDriveNo : [in] SD カード番号を指定します。(VT_I4)
 iSettingNo : [in] 検査設定 No.を指定します。(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
call caoCtrl.Execute("ChangeInspectSettingAsync", Array(1, 1))
' ChangeInspectionSettingAsync コマンドの戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")
vntResult : 戻り値 (Empty)
```

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

検査設定 No.を取得します。

書式

ReadInspectSetting ()

引数 : [in] なし
 戻り値 : [out] < iDriveNo >, < iSettingNo > (VT_I4 | VT_ARRAY)

iDriveNo : SD カード番号

iSettingNo : 検査設定 No

現在設定されている検査 No の取得方法を下記に示します。

使用例

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

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

3.4. 撮像制御

3.4.1. CaoController::Execute ("ChangeLightVolume") コマンド

指定した照明のボリュームを変更します。

書式 ChangeLightVolume(<ILightNo>, <IVolume> [, <ICnt> [, <INum>]])

ILightNo	:	[in] 照明 No. (VT_I4)
IVolume	:	[in] 照明ボリューム値 (VT_I4)
ICnt	:	[in] 撮像回 または撮像箇所 (VT_I4)
INum	:	[in] 複数回撮像時の照明 (VT_I4)
戻り値	:	[out] なし

照明 1 のボリュームを 30 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeLightVolume", Array(1, 30)
```

3.4.2. CaoController::Execute ("SetShutterSpeed") コマンド

指定したカメラのシャッタースピードを変更します。

書式 SetShutterSpeed(<ICameraNo>, <IShutterSpeed> [, <ICnt> [, <INum>]])

ICameraNo	:	[in] カメラ No. (VT_I4)
IShutterSpeed	:	[in] シャッタースピード (VT_I4)
ICnt	:	[in] 撮像回 または撮像箇所 (VT_I4)
INum	:	[in] 複数回撮像時の照明 (VT_I4)
戻り値	:	[out] なし

カメラ 1 のシャッタースピードを 5 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "SetShutterSpeed", Array(1, 5)
```

3.4.3. CaoController::Execute ("SetSensitivity") コマンド

指定したカメラの感度を変更します。

書式 SetSensitivity(<ICameraNo>, <ISensitivity> [, <ICnt> [, <INum>]])

ICameraNo : [in] カメラ No. (VT_I4)
ISensitivity : [in] カメラ感度 (VT_I4)
ICnt : [in] 撮像回 または撮像箇所 (VT_I4)
INum : [in] 複数回撮像時の照明 (VT_I4)
戻り値 : [out] なし

カメラ 1 のカメラ感度を 50 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "SetSensitivity", Array(1, 50)
```

3.4.4. CaoController::Execute ("SetTriggerDelay") コマンド

トリガー入力に対して、実際に撮像が開始するまでのディレイ時間を設定します。

書式 SetTriggerDelay(<ICameraNo>, <ITriggerDelay> [, <ICnt> [, <INum>]])

ICameraNo : [in] カメラ No. (VT_I4)
ITriggerDelay : [in] トリガーディレイ (msec) (VT_I4)
ICnt : [in] 撮像回 または撮像箇所 (VT_I4)
INum : [in] 複数回撮像時の照明 (VT_I4)
戻り値 : [out] なし

カメラ 1 のトリガーディレイを 500 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "SetTriggerDelay", Array(1, 500)
```

3.5. 計測

3.5.1. CaoController::Execute (“ChangeExecuteCondition”) コマンド

現在有効な実行条件番号を, 指定した条件番号に変更します.

書式 ChangeExecuteCondition (< INo >)

INo : [in] 実行条件番号(VT_I4)

戻り値 : [out] なし

実行条件番号を 1 に設定する例を以下に示します.

使用例

```
caoCtrl.Execute “ChangeExecuteCondition”, 1
```

3.5.2. CaoController::Execute (“ReadExecuteCondition”) コマンド

現在有効な実行条件番号を読み出します.

書式 ReadExecuteCondition()

引数 : [in] なし

戻り値 : [out] 実行条件番号 (VT_I4)

例を以下に示します.

使用例

```
Dim IParam as Long
IParam = caoCtrl.Execute (“ReadExecuteCondition”)
```

3.5.3. CaoController::Execute (“WriteCharReg”) コマンド

ツール番号 nnn 番の OCR ツールの判定文字列, 1D コードリーダーツールおよび 2D コードリーダーツールの照合パターン文字列を, 指定した文字列 ssss に書き換えます. 判定文字列 ssss を指定しない場合は, そのツールの最新の読み取り結果を設定します.

書式 WriteCharReg (< IToolNo >, < IRowNo >[, < bstrParam >])

IToolNo : [in] ツール番号 (VT_I4)

IRowNo : [in] 行番号/照合条件番号 (VT_I4)

bstrParam : [in] 判定文字列 (VT_BSTR)

戻り値 : [out] なし

ツール番号 101 の OCR ツールの判定文字列を"DEF"に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "WriteCharReg", Array(101, 1, "DEF")
```

3.5.4. CaoController::Execute ("ReadCharReg") コマンド

ツール番号 nnn 番の OCR ツールの判定文字列, 1D コードリーダーツールおよび 2D コードリーダーツールの照合パターン文字列を読み出します。

書式 ReadCharReg (< IToolNo >, < IRowNo >)

IToolNo : [in] ツール番号 (VT_I4)
 IRowNo : [in] 行番号/照合条件番号 (VT_I4)
 戻り値 : [out] 判定文字列 (VT_BSTR)

ツール番号 101 の OCR ツールの判定文字列を取得する例を以下に示します。

使用例

```
Dim strParam as String  
strParam = caoCtrl.Execute("ReadCharReg", Array(101, 1))
```

3.5.5. CaoController::Execute ("ChangeToolParameter") コマンド

指定したツールの判定条件の上限値と下限値を書き換えます。

書式 ChangeToolParameter (< IToolNo >, < IID >, < IFlag >, < vntParam >)

IToolNo : [in] ツール番号 (VT_I4)
 IID : [in] 判定条件種別の項目 ID (VT_I4)
 IFlag [in] 上限(0)/下限(1)指定 (VT_I4)
 vntParam [in] 判定条件の値 (VT_VARIANT)
 戻り値 : [out] なし

ツール番号 100 のエッジツールの下限値を 123.456 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeToolParameter", Array(100, 82, 1, "123.456")
```

3.5.6. GaoController::Execute (“ReadToolParameter”) コマンド

指定したツールの判定条件の上限値と下限値を読み出します。

書式	ReadToolParameter (<IToolNo >, <IID>, <lFlag>)
IToolNo	: [in] ツール番号 (VT_I4)
IID	: [in] 判定条件種別の項目 ID (VT_I4)
lFlag	[in] 上限(0)/下限(1)指定 (VT_I4)
戻り値	: [out] 判定条件の値 (VT_BSTR)

ツール番号 100 のエッジツールの下限値を読み出す例を以下に示します。

使用例

```
Dim strParam as String
strParam = caoCtrl.Execute("ReadToolParameter", Array(100, 82, 1))
```

3.5.7. GaoController::Execute (“ChangeFlawLevel”) コマンド

指定した傷ツールの傷レベルを書き換えます。

書式	ChangeFlawLevel (<IToolNo >, <vntParam>)
IToolNo	: [in] ツール番号 (VT_I4)
vntParam	: [in] 傷レベルの値 (VT_I4)
戻り値	: [out] なし

ツール番号 102 の傷レベルを 200 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeFlawLevel", Array(102, 200)
```

3.5.8. GaoController::Execute (“ReadFlawLevel”) コマンド

指定した傷ツールの傷レベルを読み出します。

書式	ReadFlawLevel (<IToolNo >)
IToolNo	: [in] ツール番号 (VT_I4)
戻り値	: [out] 傷レベルの値 (VT_I4)

ツール番号 102 の傷レベルを取得する例を以下に示します。

使用例

```
Dim IParam as Long
IParam = caoCtrl.Execute("ReadFlawLevel", 102)
```

3.5.9. CaoController::Execute ("EnterImg") コマンド

最新の入力画像を番号 IImageNo の基準画像として保存し、保存した基準画像で基準値を計算します。

書式

EnterImg([<ICameraNo>, <IImageNo> [, <ICnt> [, <INum>]])

ICameraNo : [in] カメラ No. (VT_I4)
 IImageNo : [in] 基準画像 No. (VT_I4)
 ICnt : [in] 撮像回 または撮像箇所 (VT_I4)
 INum : [in] 複数回撮像時の照明 (VT_I4)
 戻り値 : [out] なし

カメラ 1 の最新の入力画像を番号 500 の基準画像に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "EnterImg", Array(1, 500)
```

3.5.10. CaoController::Execute ("UpdateCapturePos") コマンド

すべてのカメラまたは指定したカメラ番号の、ロボットビジョンツールの撮像位置座標を更新します。

書式

UpdateCapturePos(<ICameraNo>, <vntPosX>, <vntPosY>, <vntHeight>, <vntAngleX>, <vntAngleY>, <vntAngleZ>)

ICameraNo : [in] カメラ No. (VT_I4)
 vntPosX : [in] X 位置(VT_VARIANT)
 vntPosY : [in] Y 位置(VT_VARIANT)
 vntHeight : [in] 高さ (VT_VARIANT)
 vntAngleX : [in] 角度 X(VT_VARIANT)
 vntAngleY : [in] 角度 Y(VT_VARIANT)
 vntAngleZ : [in] 角度 Z(VT_VARIANT)

戻り値 : [out] なし

カメラ 1 の最新の撮像位置座標を更新する例を以下に示します。

使用例

```
caoCtrl.Execute "UpdateCapturePos", Array(1, 500, 600, 700, 0, 60, 120)
```

3.5.11. CaoController::Execute ("RegisterOneCharToLibrary") コマンド

OCR ツールまたは OCR2 ツールで読み取った文字を辞書に登録します。

書式

RegisterOneCharToLibrary(<IToolNo>, <IRowNo>, <ICharNo>, <ICharType>)

IToolNo : [in] ツール番号 (VT_I4)

IRowNo : [in] 検出結果行番号 (VT_I4)

ICharNo : [in] 検出結果文字番号 (VT_I4)

ICharType : [in] 登録先文字種 (VT_I4)

戻り値 : [out] なし

ツール番号 100 の OCR ツールで 1 行目の 1 文字目に読み取った文字を文字種"2"として辞書登録する例を以下に示します。

使用例

```
caoCtrl.Execute "RegisterOneCharToLibrary", Array(100, 1, 1, 2)
```

3.5.12. CaoController::Execute ("DeleteOneCharFromLibrary") コマンド

指定された文字種の最後の登録番号の文字を辞書から削除します。

書式

DeleteOneCharFromLibrary(<IToolNo>, <ICharType>)

IToolNo : [in] ツール番号 (VT_I4)

ICharType : [in] 削除対象文字種 (VT_I4)

戻り値 : [out] なし

ツール番号 100 の OCR ツールで文字種"2"として辞書登録されている最後の登録番号の文字を削除する例を以下に示します。

使用例

```
caoCtrl.Execute "DeleteOneCharFromLibrary", Array(100, 2)
```

3.6. 計測値補正

3.6.1. CaoController::Execute ("ChangePreCorrectionMeasuredValue") コマンド

任意の値の補正前計測値を計算します。

書式

ChangePreCorrectionMeasuredValue(<IToolNo>, <INo>, <fParam>)

IToolNo : [in] ツール番号 (VT_I4)
 INo : [in] 計測項目番号 (VT_I4)
 fParam : [in] 数値 (VT_R4)
 戻り値 : [out] 補正值 (VT_R4)

ツール番号 101 の高さ計測ツールで計測項目 1 の条件を用いて任意の値”5”の補正前の値を取得する例を以下に示します。

使用例

```
Dim fParam as Single
fParam = caoCtrl.Execute "ChangePreCorrectionMeasuredValue", Array(101, 1, 1, 2)
```

3.6.2. CaoController::Execute ("WriteMeasuredValueCorrection") コマンド

補正值設定を書き換えて、補正值を計算します。

計算方法は 4 種類あります。下記に従い、計算方法に応じたパラメータを指定してください。

- ① 1 点補正で補正前後の値を指定して補正します

書式

WriteMeasuredValueCorrection(<IToolNo>, <bstrNo>, <IType>, <fBeforeValue>, <fAfterValue>)

IToolNo : [in] ツール番号 (VT_I4)
 bstrNo : [in] 計測項目番号 (VT_BSTR)
 ※全ての計測項目を指定する場合は, ”AL”
 IType : [in] 補正方法 (VT_I4) ※0: 1 点補正
 fBeforeValue : [in] 補正前の値 (VT_R4)
 fAfterValue : [in] 補正後の値 (VT_R4)

戻り値 : [out] なし

- ② 1点補正で補正後の値から指定したオフセット値を差し引いて補正值とします

書式

WriteMeasuredValueCorrection(<lToolNo>, <bstrNo>, <lType>, <fOffset>)

lToolNo : [in] ツール番号 (VT_I4)
 bstrNo : [in] 計測項目番号 (VT_BSTR)
 ※全ての計測項目を指定する場合は, "AL"
 lType : [in] 補正方法 (VT_I4) ※0: 1点補正
 fOffset : [in] オフセット値 (VT_R4)
 戻り値 : [out] なし

- ③ 2点補正で補正 1 および 2 のそれぞれの補正前後の値を指定して補正します

書式

WriteMeasuredValueCorrection(<lToolNo>, <bstrNo>, <lType>, <fBeforeValue1>, <fAfterValue1>, <fBeforeValue2>, <fAfterValue2>)

lToolNo : [in] ツール番号 (VT_I4)
 bstrNo : [in] 計測項目番号 (VT_BSTR)
 ※全ての計測項目を指定する場合は, "AL"
 lType : [in] 補正方法 (VT_I4) ※1: 2点補正
 fBeforeValue1 : [in] 補正前の値 1 (VT_R4)
 fAfterValue1 : [in] 補正後の値 1 (VT_R4)
 fBeforeValue2 : [in] 補正前の値 2 (VT_R4)
 fAfterValue2 : [in] 補正後の値 2 (VT_R4)
 戻り値 : [out] なし

- ④ 2点補正で補正後の値から指定した係数 A および係数 B を逆算して補正值とします

書式

WriteMeasuredValueCorrection(<lToolNo>, <bstrNo>, <lType>, <fCoefficientA>, <fCoefficientB>)

lToolNo : [in] ツール番号 (VT_I4)
 bstrNo : [in] 計測項目番号 (VT_BSTR)
 ※全ての計測項目を指定する場合は, "AL"
 lType : [in] 補正方法 (VT_I4) ※0: 1点補正
 fCoefficientA : [in] 係数 A (VT_R4)
 fCoefficientB : [in] 係数 B (VT_R4)
 戻り値 : [out] なし

ツール番号 101 の高さ計測ツールで計測項目 1 に①の補正值設定を行う例を以下に示します。

使用例

```
' 1点補正 補正前"2.0" → 補正後"6.0"
caoCtrl.Execute "WriteMeasuredValueCorrection", Array(101, "1", 0, 2.0, 6.0)
```

3.6.3. CaoController::Execute("ReadMeasuredValueCorrection") コマンド

指定したツールで設定している計測値補正の値を返します。

書式

ReadMeasuredValueCorrection(<IToolNo>, <INo>, <fParam>)

IToolNo : [in] ツール番号 (VT_I4)

INo : [in] 計測項目番号 (VT_I4)

戻り値 : [out] (VT_I4 | VT_ARRAY)

① 1点補正の場合

0, <fBeforeValue>, <fAfterValue>, <fOffset>

fBeforeValue : 補正前の値 (VT_R4)

fAfterValue : 補正後の値 (VT_R4)

fOffset : 補正前の値 (VT_R4)

② 2点補正の場合

1, <fBeforeValue1>, <fAfterValue1>, <fBeforeValue2>, <fAfterValue2>, <fCoefficientA>, <fCoefficientB>

fBeforeValue1 : 補正前の値 1 (VT_R4)

fAfterValue1 : 補正後の値 1 (VT_R4)

fBeforeValue2 : 補正前の値 2 (VT_R4)

fAfterValue2 : 補正後の値 2 (VT_R4)

fCoefficientA : 係数 A (VT_R4)

fCoefficientB : 係数 B (VT_R4)

ツール番号 101 の高さ計測ツールで計測項目 1 から補正值設定する例を以下に示します。

使用例

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute "ReadMeasuredValueCorrection", Array(101, 1)
```

3.7. 独自コマンド

3.7.1. GaoController::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、CVX シリーズのユーザーズマニュアルを参照してください。

書式 ExecuteCommand(< bstrCommand >)

bstrCommand : [in] コマンドの文字列を指定します (VT_BSTR)

戻り値 : [out] コマンドの応答を返します。 (VT_BSTR)

無手順コマンドを指定し、CVX 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 strRet as String
strRet = caoCtrl.Execute("ExecuteCommand", "R0")
```

3.7.2. GaoController::Execute (“ExecuteCommandAsync”) コマンド (非推奨)

指定した無手順コマンドを非同期で実行します。

サポートする無手順コマンドについては、CVX シリーズのユーザーズ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、3.7.8.CaoController::Execute (“GetCommandResult”) コマンドを参照ください。

コマンド使用時は、必ず GetCommandResult コマンドを実行後にプロバイダを切断してください。

非同期処理で異常が発生する場合があります。

書式 ExecuteCommandAsync(< bstrCommand >)

bstrCommand : [in] コマンドの文字列を指定します (VT_BSTR)

戻り値 : [out] なし

無手順コマンドを指定し、CVX 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

Call 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)
```

3.7.3. CaoController::Execute (“TriggerAndGetResult”) コマンド

トリガをかけた後に結果の受信を行います。出力結果が受信できない場合は、Timeout 時間まで待ちます。

書式 TriggerAndGetResult(< iTriggerNo >)

iTriggerNo : [in] トリガ番号 1~4 (VT_I2)

戻り値 : [out] 出力結果 (VT_BSTR)

例を以下に示します。

使用例

```
Dim strRet as String
strRet = caoCtrl.Execute("TriggerAndGetResult", 1)
```

3.7.4. CaoController::Execute (“RecievePacket”) 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 RecievePacket ()

引数 : [in] なし

戻り値 : [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
caoCtrl.Execute "Trigger", 1
strRet = caoCtrl.Execute("RecievePacket")
```

3.7.5. CaoController::Execute (“ClearPacket”) コマンド

受信バッファにあるパケットを消去します。

書式 ClearPacket()

引数 : なし

戻り値 : [out] なし

例を以下に示します。

使用例

```
caoCtrl.Execute "ClearPacket"
```

3.7.6. CaoController::Execute ("SetTimeout") コマンド

タイムアウト時間を設定します

書式 SetTimeout(<iTimeout>)

iTimeout : [in] タイムアウト時間 (msec) (VT_UI4)
戻り値 : [out] なし

使用例を示します.

使用例

```
caoCtrl.Execute "SetTimeout", 1000
```

3.7.7. CaoController::Execute ("GetTimeout") コマンド

設定されているタイムアウト時間を取得します.

書式 GetTimeout()

引数 : [in] なし
戻り値 : [out] タイムアウト値 (msec) (VT_UI4)

例を以下に示します.

使用例

```
Dim timeout as Long  
timeout = caoCtrl.Execute("GetTimeout")
```

3.7.8. 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 (0x80100003)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
Call caoCtrl.Execute("ExecuteCommandAsync", "R0")
vntResult = caoCtrl.Execute("GetCommandResult")
```