

CV プロバイダ KEYENCE CV シリーズ用

Version 1.0.2

ユーザーズガイド

August 23, 2019

【備考】

【改版履歴】

版数	日付	内容
1.0.0	2014-02-11	初版
1.0.1	2016-01-13	Trigger コマンド引数拡張 TriggerAndGetResult コマンドバグ修正 非同期コマンド (ChangeModeAsync, ChangeInspectionSettingAsync, RefreshReferencePositionAsync) 追加 GetCommandResult コマンド追加 エラーコード一覧追加
1.0.2	2019-03-18	同期コマンド (RegistImage, GetRegistImageList, SaveAllImage, GetLibraryList, ChangePassword, InputPseudoConsole, SaveStatisticalData, ChangeShutterSpeed, ChangeCameraSensitivity, ChangeTriggerDelay, ChangeLightIntensityLevel, ChangePatternCounter)追加
1.0.2	2019-08-23	非同期コマンド使用時の注意点追加

目次

1. はじめに	5
1.1. 機器の設定.....	5
1.1.1. RS232C 設定	5
1.1.2. Ethernet 設定	6
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	12
3.1. トリガー.....	14
3.1.1. CaoController::Execute (“Trigger”) コマンド.....	14
3.2. コントローラ制御	15
3.2.1. CaoController::Execute (“ChangeMode”) コマンド.....	15
3.2.2. CaoController::Execute (“ChangeModeAsync”) コマンド (非推奨).....	15
3.2.3. CaoController::Execute (“Reset”) コマンド	16
3.2.4. CaoController::Execute (“StoreSetting”) コマンド.....	16
3.2.5. CaoController::Execute (“ChangeDisplayPattern”) コマンド	16
3.2.6. CaoController::Execute (“ChangeCurrentUnit”) コマンド.....	17
3.2.7. CaoController::Execute (“ReadCurrentUnit”) コマンド	17
3.2.8. CaoController::Execute (“ReoutputResult”) コマンド.....	17
3.2.9. CaoController::Execute (“RegistImage”) コマンド	18
3.2.10. CaoController::Execute (“GetRegistImageList”) コマンド	18
3.2.11. CaoController::Execute (“SaveAllImage”) コマンド	19
3.2.12. CaoController::Execute (“GetLibraryList”) コマンド.....	19
3.2.13. CaoController::Execute (“ChangePassword”) コマンド	20
3.2.14. CaoController::Execute (“InputPseudoConsole”) コマンド	21
3.2.15. CaoController::Execute (“SaveStatisticalData”) コマンド.....	22
3.3. 検査設定切替	22
3.3.1. CaoController::Execute (“ChangeInspectSetting”) コマンド	22

3.3.2. CaoController::Execute ("ChangeInspectSettingAsync") コマンド (非推奨)	23
3.3.3. CaoController::Execute ("ReadInspectSetting") コマンド	23
3.3.4. CaoController::Execute ("ReadToolParameter") コマンド	24
3.3.5. CaoController::Execute ("ChangeToolParameter") コマンド	24
3.3.6. CaoController::Execute ("ReadBinaryData") コマンド	25
3.3.7. CaoController::Execute ("ChangeBinaryData") コマンド	25
3.3.8. CaoController::Execute ("InitCommandMemory") コマンド	26
3.3.9. CaoController::Execute ("ReadCommandMemory") コマンド	26
3.3.10. CaoController::Execute ("ChangeCommandMemory") コマンド	26
3.3.11. CaoController::Execute ("RefreshReferencePosition") コマンド	27
3.3.12. CaoController::Execute ("RefreshReferencePositionAsync") コマンド (非推奨)	27
3.4. 入出力制御	28
3.4.1. CaoController::Execute ("EnableTrigger") コマンド	28
3.5. 撮像制御	28
3.5.1. CaoController::Execute ("ChangeShutterSpeed") コマンド	28
3.5.2. CaoController::Execute ("ChangeCameraSensitivity") コマンド	29
3.5.3. CaoController::Execute ("ChangeTriggerDelay") コマンド	29
3.5.4. CaoController::Execute ("ChangeLightIntensityLevel") コマンド	30
3.5.5. CaoController::Execute ("ChangePatternCounter") コマンド	30
3.6. 独自コマンド	30
3.6.1. CaoController::Execute ("ExecuteCommand") コマンド	30
3.6.2. CaoController::Execute ("ExecuteCommandAsync") コマンド (非推奨)	31
3.6.3. CaoController::Execute ("TriggerAndGetResult") コマンド	31
3.6.4. CaoController::Execute ("RecievePacket") コマンド	32
3.6.5. CaoController::Execute ("ClearPacket") コマンド	32
3.6.6. CaoController::Execute ("SetTimeout") コマンド	33
3.6.7. CaoController::Execute ("GetTimeout") コマンド	33
3.6.8. CaoController::Execute ("GetCommandResult") コマンド	33

1. はじめに

本書は KEYENCE 社製ビジョンシステムである CV シリーズ用の CAO プロバイダである、CV プロバイダのユーザーズガイドです。

CV プロバイダは Ethernet 接続か RS232C 接続された CV 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 CV 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 RS232C か Ethernet 通信が使用できます。CV シリーズの環境設定メニューより、接続する手段を選択して設定してください。

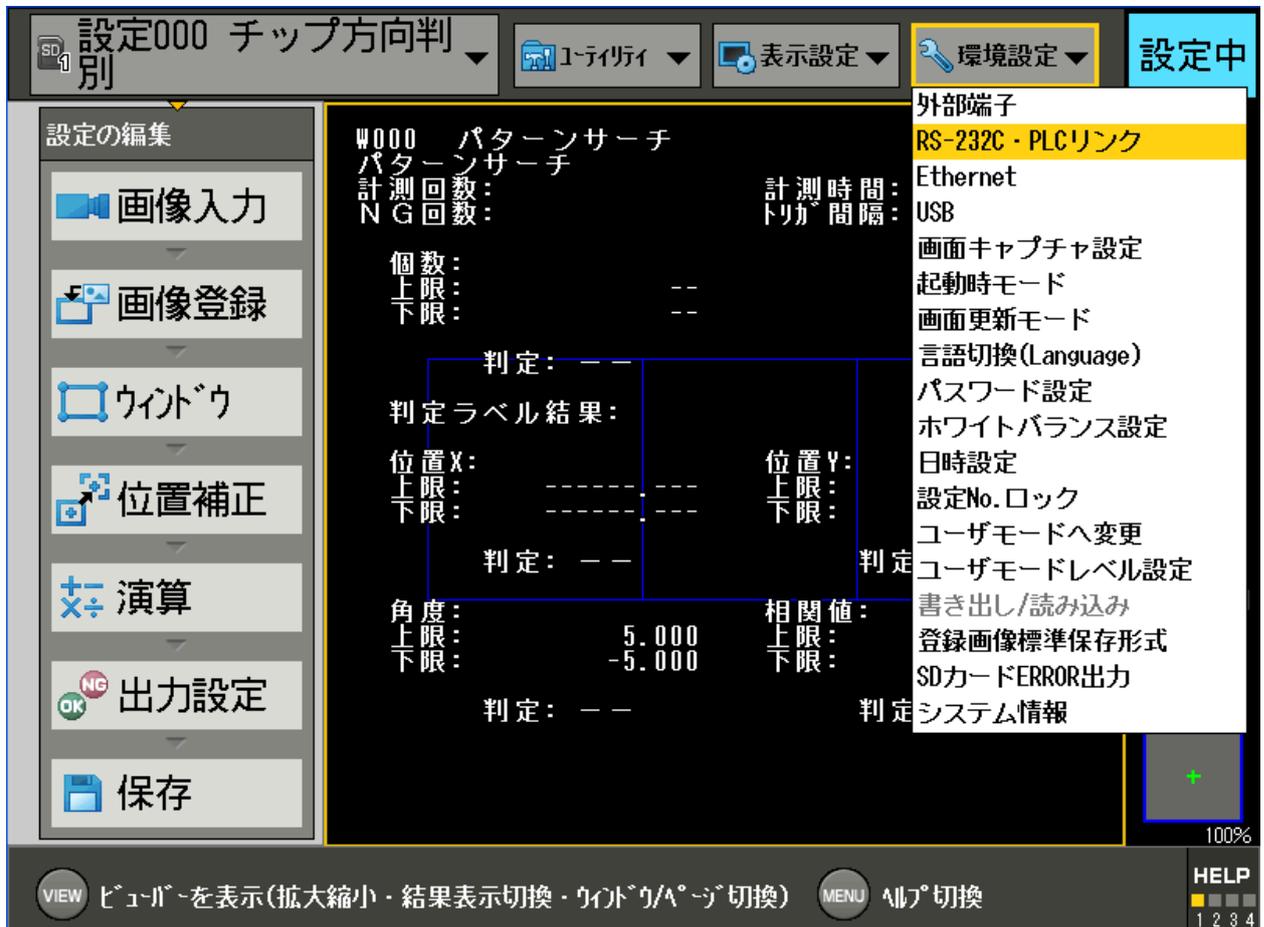


図 1-1 環境設定

1.1.1. RS232C 設定

RS232C 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。

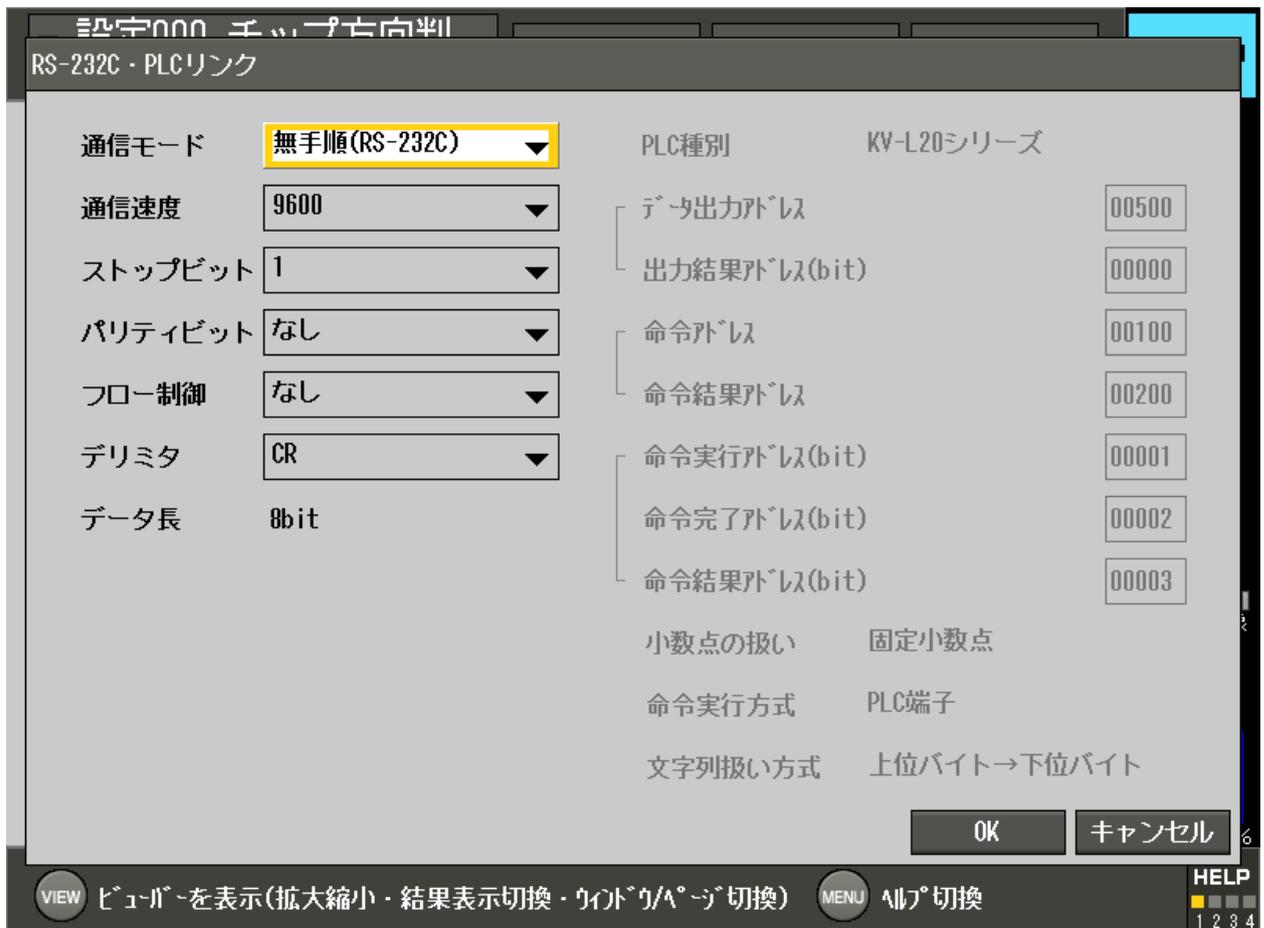


図 1-2 RS232C 設定

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。

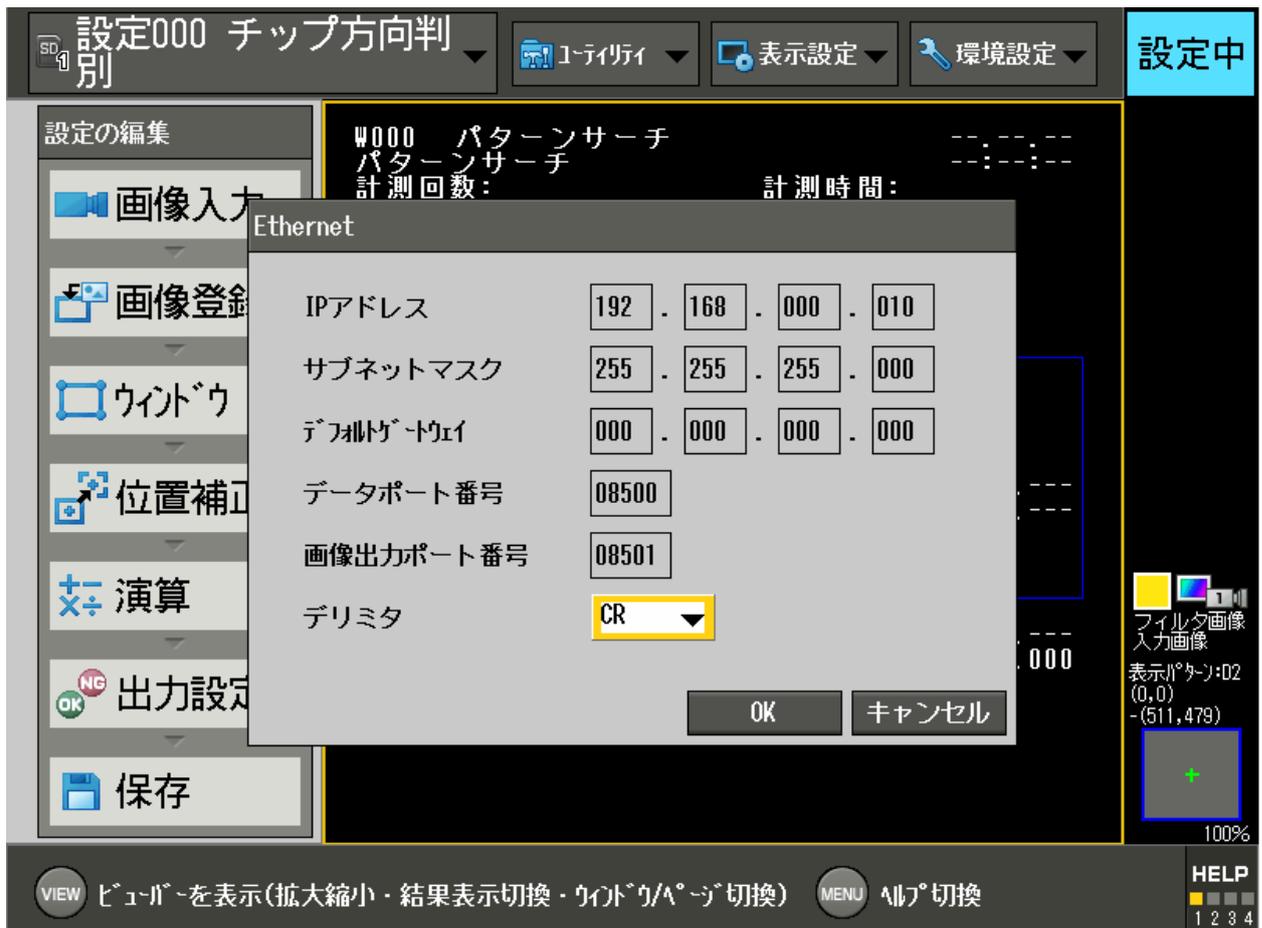


図 1-3 Ethernet 設定

2. プロバイダの概要

2.1. 概要

CV 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 CV プロバイダ

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

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

CV 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.CV”
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:

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。

500)

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 CV 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("CV", "caoProv.KEYENCE.CV", "",
"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("CV", "caoProv.KEYENCE.CV", "",
"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 CV シリーズからの処理結果は HRESULT として返されます。CV 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 CV 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 CVエラー	CV 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については KEYENCE 社の CV 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

CVシリーズ 無手順コマンド	コマンド	機能	
トリガー			
T1, T2	Trigger	トリガを入力します	P. 14
コントローラ制御			
R0, S0	ChangeMode	モードを移行します	P. 15
R0, S0	ChangeModeAsync	モードを非同期で移行します (非推奨)	P. 15
RS	Reset	各種リセットを行います	P. 16
SS	StoreSetting	設定を保存します	P. 16
DS	ChangeDisplayPattern	表示パターンを切り換えます	P. 16
UW	ChangeCurrentUnit	ウィンドウ No を切り替えます	P. 17
UR	ReadCurrentUnit	ウィンドウNoを読み出します	P. 17
M0	ReoutputResult	最新の処理結果を再出力します	P. 17
BS	RegistImage	画像を登録します	P. 18
BL	GetRegistImageList	登録画像の一覧を取得します	P. 18
BT	SaveAllImage	全画像データを読み出します	P. 19
DL	GetLibraryList	辞書の一覧を取得します	P. 19
PS	ChangePassword	パスワードを変更します	P. 20
KY	InputPseudoConsole	コンソール疑似入力します	P. 21
ST	SaveStatisticalData	統計データを保存する	P. 22
検査設定切替			
PW	ChangeInspectSetting	設定 No.を切り換えます	P. 22
PW	ChangeInspectSettingAsync	設定 No.を非同期で切り替えます (非推奨)	P. 23
PR	ReadInspectSetting	設定 No.を読み出します	P. 23
DR	ReadToolParameter	公差を読み出します	P. 24
DW	ChangeToolParameter	公差を書き込みます	P. 24
JR	ReadBinaryData	2 値化データを読み出します	P. 25
JW	ChangeBinaryData	2 値化データを書き込みます	P. 25
MI	InitCommandMemory	コマンドメモリの初期値を設定します	P. 26

MR	ReadCommandMemory	コマンドメモリのデータを読み出します	P. 26
MW	ChangeCommandMemory	コマンドメモリにデータを書き込みます	P. 26
RR	RefreshReferencePosition	基準値を再計算します	P. 27
RR	RefreshReferencePositionAsync	非同期で基準値を再計算します (非推奨)	P. 27
入出力制御			
TE	EnableTrigger	トリガー入力禁止/許可の変更を行います	P. 28
撮像制御			
CW	ChangeShutterSpeed	シャッタースピードを変更します	P. 28
CW	ChangeCameraSensitivity	カメラの感度を変更します	P. 29
CW	ChangeTriggerDelay	撮像時のデレイ時間を設定します	P. 29
CW	ChangeLightIntensityLevel	照明のボリュームを変更します	P. 30
CW	ChangePatternCounter	照明点灯パターンを変更します	P. 30
独自拡張コマンド			
-	ExecuteCommand	無手順コマンドを実行します	P. 30
-	ExecuteCommandAsync	非同期で無手順コマンドを実行します (非推奨)	P. 31
-	TriggerAndGetResult	トリガを発行し、出力結果を受信します	P. 31
-	RecievePacket	パケットの受信を行います	P. 32
-	ClearPacket	バッファの受信パケットを破棄します	P. 32
-	GetTimeout	タイムアウト時間の取得を行います	P. 33
-	SetTimeout	タイムアウト時間の設定を行います	P. 33
-	GetCommandResult	非同期コマンドの戻り値を取得します	P. 33

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。

書式

Trigger(< iTriggerNo > , [<iMode>])

iTriggerNo : [in] 発行対象のトリガ番号を指定します。(VT_I4)

1: トリガ1

2: トリガ2

iMode : [in] 結果を受信するかどうかのフラグを設定します。(VT_I4)

0: トリガを発行し、応答結果を受信しません。

1: トリガを発行し、応答結果を受信します。(デフォルト)

戻り値 : [out] 結果文字列 (iMode=1 の時)

結果を後で受信する際には先頭にトリガコマンドの応答パケット「T1,」が付与されています。エラーの際には応答パケットが「ER,T1,nn」となります。nn には CV のエラーコードが格納されます。

また、結果を受信するまで、他のコマンドを使用しないでください。

トリガ番号1を発行し、x=11,y=12 の値を出力する例を以下に示します。

使用例

```
Dim strResult as String
' 結果を受信する時
' トリガ発行&結果取得
strResult = caoCtrl.Execute("Trigger", 1)
' strResult: "+11,+12"

' 結果を後で受信する時
' トリガだけ発行
Call caoCtrl.Execute("Trigger", Array(1,0))
' 何かの処理
' 結果取得
strResult = caoCtrl.Execute("RecievePacket")
' strResult: "T1,+11,+12"
```

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
 0: 停止モード
 1: 運転モード

戻り値 : [out] 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、3.6.8.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
 0: 停止モード
 1: 運転モード

戻り値 : [out] 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
' ChangeMoveAsync 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 : [in] なし

戻り値 : [out] なし

例を以下に示します。

使用例

```
caoCtrl.Execute “Reset”
```

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

現在選ばれている設定 No.の設定値を保存します。

書式

StoreSetting

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 “StoreSetting”
```

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

表示設定の切り替えを行います。

書式

ChangeDisplayPattern(<bstrMode>, <bstrParam>)

bstrMode : [in] 切り替え対象 (VT_BSTR)

PT : 表示パターン

RS : 結果表示

PG : ページ

FC : 画面

bstrParam : [in] 上記パラメータに応じたパラメータ (VT_BSTR)

戻り値 : なし

ページを切り換える際に前のページに移動する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeDisplayPattern", Array("PG", "P")
```

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

指定したウィンドウ No に切り替えます。

書式 ChangeCurrentUnit (<iWindowNo>)

iWindowNo : [in] ウィンドウ No (0~127) (VT_I4)

戻り値 : [out] なし

ウィンドウ No2に切り替える例を下記に示します。

使用例

```
Call caoCtrl.Execute("ChangeCurrentUnit", 2)
```

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

現在のウィンドウ No を返します。

書式 ReadCurrentUnit()

引数 : [in] なし

戻り値 : [out] ウィンドウ No (0~127) (VT_I4)

現在のウィンドウ No を取得する例を下記に示します。

使用例

```
Din INum as Long  
INum = caoCtrl.Execute("ReadCurrentUnit")
```

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

最新の計測結果を再読み出しします。

書式 ReoutputResult ()

引数 : [in] なし

戻り値 : [out] 計測結果 (VT_BSTR)

例を下記に示します。

使用例

```
Din bstrResult as String
bstrResult = caoCtrl.Execute("ReoutputResult")
```

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

カメラ No.で指定された最新の取り込み画像を、指定した No.の登録画像として登録します。

書式

RegistImage (<ICameraNo>, <IRegistImageNo>)

ICameraNo : [in] カメラ No (VT_I4)
1~4

IRegistImageNo : [in] 登録画像 No (VT_I4)
0~999

戻り値 : なし

カメラ No.1 に最新の取り込み画像を登録画像 No.0 で登録する例を下記に示します。

使用例

```
Call caoCtrl.Execute("RegistImage", Array(1,0))
```

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

設定されている登録画像番号を一覧で取得します。

1. 現在の設定の登録画像の一覧を取得する場合

書式

GetRegistImageList (<ITarget>, <ICameraNo>)

ITarget : [in] 取得先(VT_I4)
0:現在の設定

ICameraNo [in] カメラ No(VT_I4)
1~4

戻り値 : [out]登録画像番号一覧(VT_I4 | VT_ARRAY)

2. SD カードに保存されている登録画像の一覧を取得する場合

書式

GetRegistImageList (<ITarget>, <IRegistImageNo>,<ICameraNo>)

ITarget : [in] 取得先(VT_I4)
1:SD カード 1
2:SD カード 2

IRegistImageNo : [in] 登録画像の設定の番号(VT_I4)
0~999

1. 現在の設定 No.の辞書番号の一覧を取得する場合



GetLibraryList (<ITarget>)

ITarget : [in] 取得先(VT_I4)

0:現在の設定

戻り値 : [out]辞書番号一覧(VT_I4 | VT_ARRAY)

2. SD カードに保存されている辞書番号の一覧を取得する場合



GetLibraryList (<ITarget>, <ISettingNo>)

ITarget : [in] 取得先(VT_I4)

1:SD カード 1

2:SD カード 2

ISettingNo : [in] 取得する辞書番号の設定 No.(VT_I4)

0~999

戻り値 : [out] 辞書番号一覧(VT_I4 | VT_ARRAY)

現在の設定にある辞書番号の一覧を取得する例を下記に示します。



```
Dim vntRet as variant
vntRet = caoCtrl.Execute("GetLibraryList", 0)
```

3.2.13. GaoController::Execute ("ChangePassword") コマンド

パスワードを変更します。



ChangePassword (<IOldPassword>, <INewPassword>)

IOldPassword : [in] 旧パスワード(VT_I4)

0000~9999

INewPassword : [in] 新パスワード(VT_I4)

0000~9999

戻り値 : なし

パスワードを「0000」から「1111」に変更する例を下記に示します。



```
call caoCtrl.Execute("ChangePassword", Array("0000", "1111"))
```

3.2.14. GaoController::Execute (“InputPseudoConsole”) コマンド

疑似的にコンソール入力をおこなうことができます。



InputPseudoConsole (<bstrConsoleInputCode>)

bstrConsoleInputCode : [in] コンソール入力コード(VT_BSTR)

FN:FUNCTION ボタン

ES:ESCAPE ボタン

TG:TRG ボタン

SC:SCREEN ボタン

VI:VIEW ボタン

MN:MENU ボタン

EN:ENTER ボタン

UP:ENTER ボタン上

DN:ENTER ボタン下

LT:ENTER ボタン左

RT:ENTER ボタン右

LU:ENTER ボタン左上

LD:ENTER ボタン左下

RU:ENTER ボタン右上

RD:ENTER ボタン右下

FU:FUNCTION+ENTER 上

FD:FUNCTION+ENTER 下

FL:FUNCTION+ENTER 左

FR:FUNCTION+ENTER 右

FLU:FUNCTION+ENTER ボタン左上

FLD:FUNCTION+ENTER ボタン左下

FRU:FUNCTION+ENTER ボタン右上

FRD:FUNCTION+ENTER ボタン右下

RS:運転/設定切り換え

FV:FUNCTION+VIEW

FT:FUNCTION+TRG

FM:FUNCTION+MENU

FE:FUNCTION+ENTER

FS:FUNCTION+ESCAPE

SL:SCREEN+ENTER 左

SR:SCREEN+ENTER 右

戻り値 : なし

疑似的にコンソールで Function ボタンを押す例を下記に示します。

使用例

```
call caoCtrl.Execute("InputPseudoConsole", "FN")
```

3.2.15. CaoController::Execute ("SaveStatisticalData") コマンド

統計解析で保存されているすべての統計データを、カンマ区切りのテキスト形式で SD カードに書き込みます。

書式

SaveStatisticalData (< bstrFolderDirectory >)

bstrFolderDirectory : [in] SD カードの出力先ディレクトリ(VT_BSTR)

戻り値 : なし

統計データをフォルダ「stat」にテキスト形式で書き込みする例を以下に示します。

使用例

```
call caoCtrl.Execute("SaveStatisticalData", "/CV/stat")
```

3.3. 検査設定切替

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)

1 : SD1

2 : SD2

iSettingNo : [in] 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

```
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、3.6.8.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)

1 : SD1

2 : SD2

iSettingNo : [in] 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
call caoCtrl.Execute("ChangeInspectSettingAsync", Array(1, 1))
' ChangeInspectSettingAsync 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
```

3.3.4. CaoController::Execute (“ReadToolParameter”) コマンド

指定したウィンドウの公差の上限値と下限値を読み出します。

書式

ReadToolParameter (<bstrWindow>, <bstrKind>, <bstrLimit>)

bstrWindow : [in] 指定ウィンドウ (VT_BSTR)
 計算ウィンドウ : W000 ~ W127
 演算ウィンドウ : C000 ~ C127

bstrKind : [in] 公差種別 (VT_BSTR)
 詳しくは KEYENCE 社のユーザーズマニュアルを参照してください。

bstrLimit : [in] 上限/下限指定 (VT_BSTR)
 HL: 上限値
 LL: 下限値

dblMargin : [out] 公差設定値 (VT_R8)

計測ウィンドウ 005(パターンサーチ)の X 座標下限値(256.030)を読み出す例を以下に示します。

使用例

```
Dim dblMargin as Double
dblMargin = caoCtrl.Execute("ReadToolParameter", Array("W005", "X", "LL"))
' dblMargin : 256.030
```

3.3.5. CaoController::Execute (“ChangeToolParameter”) コマンド

指定したウィンドウの公差の上限値と下限値を設定します。

書式

ChangeToolParameter (<bstrWindow>, <bstrKind>, <bstrLimit>, <bstrMargin>)

bstrWindow : [in] 指定ウィンドウ (VT_BSTR)
 計算ウィンドウ : W000 ~ W127
 演算ウィンドウ : C000 ~ C127

bstrKind : [in] 公差種別 (VT_BSTR)
 詳しくは KEYENCE 社のユーザーズマニュアルを参照してください。

bstrLimit : [in] 上限/下限指定 (VT_BSTR)
 HL: 上限値
 LL: 下限値

bstrMargin : [in] 公差設定値 (VT_BSTR)

戻り値 : なし

演算ウィンドウ 010 の下限値を-142.214 に変更する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeToolParameter", Array("C010", "MS", "LL", "-142.214")
```

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

指定したウィンドウの 2 値化フィルタについて、上限値と下限値を読み出します。

書式

ReadBinaryData (<IWindowNo>)

IWindowNo : [in] 指定ウィンドウ No (VT_I4)
0~127

戻り値 : [out] 上限値と下限値(VT_I4 | VT_ARRAY) (max, min)
max : 2 値化データ上限(0~255)
min : 2 値化データ下限(0~255)

ウィンドウ 3 の 2 値化フィルタの上限, 下限値を取得する例を以下に示します。

使用例

```
Dim vntParams as Variant
vntParams = caoCtrl.Execute("ReadBinaryData", 3)

' vntParams(0) : 255
' vntParams(1) : 125
```

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

指定したウィンドウの 2 値化フィルタについて、上限値と下限値を設定します。

書式

ChangeBinaryData (<IWindowNo>, <IMax>, <IMin>)

IWindowNo : [in] 指定ウィンドウ No (VT_I4)
0~127

IMax : [in] 2 値化データ上限 (0~255)

IMin : [in] 2 値化データ下限 (0~255)

戻り値 : [out] なし

ウィンドウ 3 の 2 値化フィルタの上限値を 200 に, 下限値を 100 に設定する例を以下に示します。

使用例

```
caoCtrl.Execute "ChangeBinaryData", Array(3, 200, 100)
```

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

コマンドメモリの初期設定値として、現在のコマンドメモリの値をまとめて反映します。

書式 InitCommandMemory ()

引数 : [in] なし
戻り値 : [out] なし

使用例を以下に示します。

使用例

```
caoCtrl.Execute "InitCommandMemory"
```

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

指定したコマンドメモリのデータを読み出します。

書式 ReadCommandMemory (<ICommandMemNo>)

ICommandMemNo : [in] コマンドメモリ No. (VT_I4)
 0～999
戻り値 : [out] 読み出しデータ (VT_I4)
 -2147483648～2147483647

使用例を以下に示します。

使用例

```
Dim IParam as Long  
IParam = caoCtrl.Execute("ReadCommandMemory", 4)
```

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

指定した最大 32 個のコマンドメモリにデータを設定します。

書式 ChangeCommandMemory(<ICommandMemNo1>, <IParam1> [,ICommandMemNo2,
 IParam2 ..., ICommandMemNon, IParamn])

ICommandMemNon : [in] コマンドメモリ No. (VT_I4)
 0～999

IParamn : [in] 書き込みデータ (VT_I4)
-2147483648～2147483647

戻り値 : [out] なし

コマンドメモリ No と書き込みデータは必ずペアで指定して下さい。

コマンドメモリ 000 の値を 1, コマンドメモリ 100 の値を-1000 に設定する場合の例を以下に示します。

使用例

```
Dim IParam as Long
IParam = caoCtrl.Execute("ChangeCommandMemory", Array(0, 1, 100, -1000))
```

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

現在の登録画像で基準値を再計算します。

書式 RefreshReferencePosition()

引数 : [in] なし

戻り値 : [out] なし

使用例を以下に示します。

使用例

```
caoCtrl.Execute "RefreshReferencePosition"
```

3.3.12. CaoController::Execute ("RefreshReferencePositionAsync") コマンド (非推奨)

非同期で現在の登録画像で基準値を再計算します。

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、3.6.8.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 RefreshReferencePositionAsync()

引数 : [in] なし

戻り値 : [out] なし

使用例を以下に示します。

使用例

```
Dim vntResult as variant
```

```
caoCtrl.Execute "RefreshReferencePositionAsync"
' RefreshReferencePositionAsync の戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")
' vntResult : 戻り値 (Empty)
```

3.4. 入出力制御

3.4.1. 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_I4)
 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。



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

3.5. 撮像制御

3.5.1. CaoController::Execute ("ChangeShutterSpeed") コマンド

指定したカメラのシャッタースピードを変更することができます。



ChangeShutterSpeed (<ICameraNo>, <IShutterSpeed>)

ICameraNo : [in] カメラ No.(VT_I4)
 1~4
 IShutterSpeed [in] シャッタースピード(VT_I4)
 0:1/15
 1:1/30
 2:1/60
 3:1/120
 4:1/240
 5:1/500

6:1/1000
 7:1/2000
 8:1/5000
 9:1/10000
 10:1/20000

戻り値 : なし

カメラ No.1 のシャッタースピードを 1/30 に変更する例を下記に示します。

使用例

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

3.5.2. CaoController::Execute ("ChangeCameraSensitivity") コマンド

指定したカメラの感度を変更することができます。

指定した値の 1/10 の値がカメラ感度として設定されます。

書式

ChangeCameraSensitivity (<ICameraNo>, <ISensitivity>)

ICameraNo : [in] カメラ No.(VT_I4)
 1~4

ISensitivity [in] 感度(VT_I4)
 10~90

戻り値 : なし

カメラ No.1 の感度を 1.0 に変更する例を下記に示します。

使用例

```
call caoCtrl.Execute("ChangeCameraSensitivity", Array(1, 10))
```

3.5.3. CaoController::Execute ("ChangeTriggerDelay") コマンド

トリガ入力に対して、実際に撮像が開始するまでのデレイ時間を設定することができます。

書式

ChangeTriggerDelay (<ICameraNo>, <IDelay>)

ICameraNo : [in] カメラ No.(VT_I4)
 1~4

IDelay [in] デレイ時間(VT_I4)
 0~999(ms)

戻り値 : なし

カメラ No.1 のトリガ入力時のディレイ時間を 100ms に変更する例を下記に示します。

使用例

```
call caoCtrl.Execute("ChangeTriggerDelay", Array(1, 100))
```

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

指定した照明のボリュームを変更することができます。

書式

ChangeLightIntensityLevel (<ILightingNo>, <ILightingVolume>)

ILightingNo : [in] 照明 No.(VT_I4)
1~8

ILightingVolume [in] 照明ボリューム値 (VT_I4)
0~255

戻り値 : なし

照明 No.1 の照明のボリュームを 50 に変更する例を下記に示します。

使用例

```
call caoCtrl.Execute("ChangeLightIntensityLevel", Array(1, 50))
```

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

マルチパターン発光のパターンカウンタを変更し照明点灯パターンを変更することができます。

書式

ChangePatternCounter (<ICounter>)

ICounter : [in] カウンタ値 (VT_I4)
0~3

戻り値 : なし

マルチパターン発光のパターンカウンタを 1 に変更する例を下記に示します。

使用例

```
call caoCtrl.Execute("ChangePatternCounter", 1)
```

3.6. 独自コマンド

3.6.1. 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、CV 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 [<vntRet> =]ExecuteCommand(< bstrCommand >)

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

vntRet : [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, CV 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")
```

3.6.2. CaoController::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.

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, CV 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, 3.6.8.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, CV 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 の戻り値取得
vntResult = caoCtrl.Execute("GetCommandResult")
' vntResult : 無手順コマンドの戻り値 (BSTR)
```

3.6.3. 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. 出力結果にはトリガの応答パケットが含まれます.

例) x=10, y=11 の値を出力設定した時

“T1,+10,+11”

書式 <strRet> = TriggerAndGetResult(< iTriggerNo >)

iTriggerNo : [in] トリガ番号 1～2 (VT_I2)

strRet : [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)
```

3.6.4. 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

Call caoCtrl.Execute("Trigger", Array(1,0))
strRet = caoCtrl.Execute("RecievePacket")
```

3.6.5. 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

例を書きに示します。

使用例

```
Call caoCtrl.Execute("ClearPacket")
```

3.6.6. 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

```
Call caoCtrl.Execute("SetTimeout", 1000)
```

3.6.7. 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 <uiTimeout> = GetTimeout()

引数 : [in] なし

<uiTimeout> : [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")
```

3.6.8. 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  
caoCtrl.Execute "ReStartAsync"  
vntResult = caoCtrl.Execute("GetCommandResult")
```
