

In-Sight プロバイダ Cognex In-Sight 用プロバイダ

Version 1.0.5

ユーザーズ ガイド

March 20, 2024

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2011-11-24 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	2012-09-11	SendMessageAndWait のコマンド 引数追加 GetMessage コマンド追加
1.0.1	2012-09-27	GetFileList 誤植修正
1.0.2	2013-01-09	コマンド追加 SendMessageAndGetEZ, SendMessageEZ, GetEZ, SetTimeoutNM, GetTimeoutNM コマンド修正 AddController
1.0.3	2013-03-19	ResetEZ コマンド追加
	2015-09-09	エラーコード追記
1.0.4	2016-02-25	GetFilelist コマンドの修正 ExecuteCommand コマンドの追加 RecievePacket コマンドの追加
1.0.5	2024-03-20	In-Sight 接続時の安定性向上

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. CaoController::AddVariable メソッド.....	7
2.2.4. CaoVariable::put_value プロパティ.....	8
2.2.5. CaoVariable::get_value プロパティ.....	8
2.3. 変数一覧.....	8
2.3.1. コントローラクラス.....	8
2.4. エラーコード.....	9
2.5. コマンドリファレンス.....	9
2.6. 基本通信.....	12
2.6.1. CaoController::Execute(“NativeMode”) コマンド.....	12
2.7. ファイルコマンドおよびジョブコマンド.....	12
2.7.1. CaoController::Execute(“LoadFile”) コマンド.....	12
2.7.2. CaoController::Execute(“StoreFile”) コマンド.....	12
2.7.3. CaoController::Execute(“DeleteFile”) コマンド.....	13
2.7.4. CaoController::Execute(“GetFile”) コマンド.....	13
2.7.5. CaoController::Execute(“SetJob”) コマンド.....	14
2.7.6. CaoController::Execute(“StoreJob”) コマンド.....	14
2.7.7. CaoController::Execute(“DeleteJob”) コマンド.....	14
2.7.8. CaoController::Execute(“GetJob”) コマンド.....	15
2.8. 設定コマンドおよびセル値コマンド.....	15
2.8.1. CaoController::Execute(“GetValue”) コマンド.....	15
2.8.2. CaoController::Execute(“SetInteger”) コマンド.....	16
2.8.3. CaoController::Execute(“SetFloat”) コマンド.....	16
2.8.4. CaoController::Execute(“SetString”) コマンド.....	16
2.8.5. CaoController::Execute(“GetInfo”) コマンド.....	17
2.8.6. CaoController::Execute(“StoreSettings”) コマンド.....	17
2.8.7. CaoController::Execute(“SetIPLock”) コマンド.....	18

2.8.8. CaoController::Execute (“GetIPLock”) コマンド	18
2.9. 実行コマンドおよびオンラインコマンド	19
2.9.1. CaoController::Execute (“SetOnline”) コマンド	19
2.9.2. CaoController::Execute (“GetOnline”) コマンド.....	19
2.9.3. CaoController::Execute (“SetEvent”) コマンド	19
2.9.4. CaoController::Execute (“SetEventAndWait”) コマンド	20
2.9.5. CaoController::Execute (“SendMessage”) コマンド.....	20
2.10. ネイティブモードの拡張コマンド	21
2.10.1. CaoController::Execute (“GetFilelist”) コマンド.....	21
2.11. 独自拡張コマンド	21
2.11.1. CaoController::Execute (“SendMessageAndWait”) コマンド	21
2.11.2. CaoController::Execute (“GetMessage”) コマンド	22
2.11.3. CaoController::Execute (“SendMessageAndGetEZ”) コマンド	23
2.11.4. CaoController::Execute (“SendMessageEZ”) コマンド	24
2.11.5. CaoController::Execute (“GetEZ”) コマンド	24
2.11.6. CaoController::Execute (“ResetEZ”) コマンド.....	25
2.11.7. CaoController::Execute (“SetTimeoutNM”) コマンド.....	25
2.11.8. CaoController::Execute (“GetTimeoutNM”) コマンド	26
2.11.9. CaoController::Execute (“ExecuteCommand”) コマンド.....	26
2.11.10. CaoController::Execute (“RecievePacket”) コマンド.....	26
3. 画像処理結果の取得方法.....	27
3.1. ネイティブモード経由での取得.....	27
3.1.1. WriteMessage 関数の利用	27
3.1.2. スプレッドシートの値の取得	29
3.2. EasyBuilder での結果取得	30
3.2.1. EasyBuilder の設定	30
3.2.2. EasyBuilder の結果取得方法.....	32
3.2.2.1. SendMessageAndGetEZ の使用例	33
3.2.2.2. SendMessageEZ と GetEZ の使用例	33
3.2.2.3. ResetEZ と GetEZ の使用例	33
3.2.2.4. イベントでの使用例	34

1. はじめに

本書は Cognex 社製ビジョンシステム用の CAO プロバイダである, In-Sight プロバイダのユーザーズガイドです.

In-Sight プロバイダは Ethernet 接続された In-Sight シリーズとネイティブモードと DataChannel モードで接続し, ネイティブモードのコマンドの送受信や画像の取得を行います. 通信は Ethernet(TCP/IP)接続をサポートしています. ネイティブモードでは XML フォーマットでの出力をサポートしていませんので, 本プロバイダを使用する際には XML フォーマットを無効化してください.

2. プロバイダの概要

2.1. 概要

In-Sight プロバイダは、コマンドの実行方法として `CaoController::Execute` と `CaoVariable` による 2 通りの方法を提供しています。

`CaoController::Execute` はネイティブモードでの通信により、In-Sight とコマンドの送受信を行うことができます。

`CaoVariable` はネイティブモードでの通信および一部の `DataChannel` による通信により、In-Sight の値や画像の取得を行います。

2.2. メソッド・プロパティ

2.2.1. `CaoWorkspace::AddController` メソッド

In-Sight 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、タイムアウト、ユーザ名、パスワード、Telnet のポート番号を指定します。

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.Cognex.In-Sight`”
`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 2-1 `CaoWorkspace::AddController` のオプション文字列

オプション	意味
<code>Conn =<接続パラメータ></code>	必須。通信形態とその接続パラメータを設定します。
<code>User=[<ユーザ名>]</code>	In-Sight にログインする為のユーザ名を指定します。 (デフォルト: admin)
<code>Password[=<Password>]</code>	ログインする為のパスワードを指定します。(デフォルト: 無し)
<code>Timeout[=<タイムアウト時間>]</code>	送受信時のタイムアウト時間(秒)を指定します。 (デフォルト: 500)
<code>EZPort[=<ポート番号>]</code>	EasyBuilder で設定した結果を受信するポート番号を指定します。 省略時は EasyBuilder で設定した結果を受信できません。 設定方法は第 3.2 章を参照してください。

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。

- Ethernet デバイス

“eth:<IP Address>[:<Port No>]”

<IP Address> : : 必須. IP アドレスを指定します.

例:"127.0.0.1"

<Port No> : : ネイティブモードで通信する為のポート番号を指定します.

(デフォルト:23)

例:"127.0.0.1:23"

2.2.2. CaoController::Execute メソッド

ネイティブモードの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 2.5 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

Execute メソッド実行した際の In-Sight からのステータスコードの戻り値は HRESULT として返されます.

戻り値 1 : S_OK (0)

戻り値 0 ~ -9 : 0x80100010 + abs(戻り値)

例: SetEvent を実行したとき.

hr = 0x80100012 : コマンドを実行することができなかったか, システムがオフラインです.

エラー内容については Cognex 社の In-Sight Explorer Reference を参照してください.

2.2.3. CaoController::AddVariable メソッド

画像の取得やセルの値取得をするための変数を作成します. 画像の取得には表 2-2 コントローラクラスシステム変数一覧を参照してください. セルの値を取得する場合には変数名を任意文字列に, オプション文字列でセルの場所を指定してください.

書式 AddVariable(<bstrVariableName:VT_BSTR>,[<bstrOption: VT_BSTR >])

bstrVariableName : [in] 変数名

bstrOption : [in] オプション文字列 (Cell=セル番号, セルの値の取得時のみ)

使用例

```
Dim objA002 as Object
Dim bstrA002 as String
objA002 = caoCtrl.AddVariable("A002", "Cell=A002")
bstrA002 = objA002.Value

bstrA002: "1.000"
```

2.2.4. CaoVariable::put_value プロパティ

現在変数クラスでは put_value プロパティをサポートしていません。

2.2.5. CaoVariable::get_value プロパティ

画像は表 2-2 コントローラクラス システム変数一覧のフォーマットで取得できます。

セルの値は VT_BSTR で取得します。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-2 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
@BITMAP	VT_UI1 VT_ARRAY	ネイティブモード経由で画像を BITMAP フォーマットで取得します。 注意:一度 In-Sight でカメラ画像の取り込みを行わないと AddVariable できません。	○	-
@BITMAP_DC	VT_UI1 VT_ARRAY	カメラ画像を Datachannel 経由で BITMAP フォーマットで取得します。カメラ画像取得するためにはトリガによりカメラ更新がかからないと取得できません。 以下のオプション文字列が使用できます。 Screen=1(デフォルト),2,4:取得する画像のサイズを決めます。サイズは 1/Screen になります。 Port=5000(デフォルト):DataChannel のポー	○	-

		ト番号を指定します。 Timeout=500(デフォルト):DataChannel のタイムアウトを設定します。 SM8=False(デフォルト):GetValue コマンドを実行する際に SM8 を送信し、トリガをかけます。		
@RAWIMG	VT_UI1 VT_ARRAY	ネイティブモード経由で画像を RAW イメージフォーマットで取得します。 * 未実装	○	-
@RAWIMG_DC	VT_UI1 VT_ARRAY	カメラ画像を DataChannel 経由で RAW イメージフォーマットにて取得します。 * 未実装	○	-

2.4. エラーコード

In-Sight 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 2-3 独自エラーコード一覧

エラー名	エラー番号	説明
E_INSIGHTERROR	0x80100010 ～ 0x80100019	In-Sight エラー (2.2.2 参照)
E_INVALIDPACKET	0x80100020	無効なパケットを受信しました
E_CONNECTION	0x80100021	接続が切断されました
E_INVALIDPASSWORD	0x80100022	無効なユーザ名・パスワードです
E_UNPREPAREDFORIMAGE	0x80100023	画像取得の準備ができていません
E_EZERROR	0x80100024	EasyBuilder 用の結果取得ポートが指定されていません

2.5. 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 Cognex 社の In-Sight Explorer Reference の通信リファレンスを参照してください。

表 2-4 CaoController::Execute コマンド一覧

In-Sight コマンド	コマンド	機能	
基本通信			
—	NativeMode	Telnet 通信により In-Sight デバイスを制御します。	P12
ファイルコマンドおよびジョブコマンド			
Load File	LoadFile	指定したジョブをメモリからロードし、アクティブなジョブにします。	P12
Store File	StoreFile	アクティブなジョブをメモリに保存します。	P12
Read File	ReadFile	未実装	
Write File	WriteFile	未実装	
Delete File	DeleteFile	メモリからジョブを削除します。	P13
Get File	GetFile	In-Sight デバイス上のアクティブなジョブのファイル名を取得します。	P13
Set Job	SetJob	In-Sight プロセッサのメモリ内のジョブスロットからジョブをロードし、それをアクティブなジョブとします。	P14
Store Job	StoreJob	In-Sight プロセッサ上のメモリ内の指定したジョブスロットに現在のジョブを保存します。	P14
Read Job	ReadJob	未実装	
Write Job	WriteJob	未実装	
Delete Job	DeleteJob	In-Sight プロセッサのメモリ内の指定したジョブスロットからジョブを削除します。	P14
Get Job	GetJob	アクティブなジョブがロードされた In-Sight のジョブスロットを返します。	P15
ファイルコマンドおよびジョブコマンド			
Read BMP	ReadBMP	未実装	
Read Image	ReadImage	未実装	
Write BMP	WriteBMP	未実装	
Write Image	WriteImage	未実装	
設定コマンドおよびセル値コマンド			
Get Value	GetValue	指定したセルに含まれている値を返します。	P15
Set Integer	SetInteger	セル内に格納されている編集ボックスコントロールを、指定した整数値に設定します。	P16
Set Float	SetFloat	セル内に格納されている編集ボックスコントロールを、指定した浮動小数点値に設定します。	P16
Set String	SetString	セル内に格納されている編集ボックスコントロールを、指定した文字列に設定します。	P16
Get Info	GetInfo	In-Sight デバイスからのシステム情報を返します。	P17
Read Settings	ReadSettings	未実装	
Write Settings	WriteSettings	未実装	
Store Settings	StoreSettings	In-Sight デバイスの現在の設定を、メモリ内の proc.set ファイルに保存します。	P17

Set IP Lock	SetIPLock	In-Sight デバイスの IP アドレスが無断で変更されることを防ぎます。	P18
Get IP Lock	GetIPLock	IP アドレスのセキュリティ状態 (アクセス不可 / アクセス可能) を返します。	P18
実行コマンドおよびオンラインコマンド			
Set Online	SetOnline	In-Sight デバイスをオンラインまたはオフラインに設定します。	P19
Get Online	GetOnline	In-Sight プロセッサのオンライン状態を返します。	P19
Set Event	SetEvent	指定したイベント (画像の取り込みなど) にトリガをかけます。	P19
Set Event & Wait	SetEventAndWait	指定したイベント (画像の取り込みなど) にトリガをかけ、コマンドが完了してからレスポンスを返します。	P20
Reset System	ResetSystem	未実装	
Abort Execution	AortExecution	未実装	
Send Message	SendMessage	ASCII 文字列をネイティブモード接続経由で In-Sight デバイスのスプレッドシートに送信します。	P20
ネイティブモードの拡張コマンド			
Get Connections	GetConnections	未実装	
Get Expr	GetExpr	未実装	
Get Filelist	GetFilelist	In-Sight デバイスのメモリに格納されているファイルの数と各ファイル名を返します。	P21
Put Live	PutLive	未実装	
Put Portnum	PutPortnum	未実装	
Put Timeout	PutTimeout	未実装	
Put Update	PutUpdate	未実装	
Put Watch	PutWatch	未実装	
Put XML	PutXML	未実装	
独自拡張コマンド			
—	SendMessageAndWait	ソフトウェアトリガの発生と WriteMessage の受信をおこないます。	P21
—	GetMessage	WriteMessage の受信を行います。	P22
—	SendMessageAndGetEZ	ソフトウェアトリガの発生と EasyBuilder で設定した TCP 通信の結果出力の受信を行います。	P23
—	SendMessageEZ	ソフトウェアトリガの発生と GetEZ コマンドで結果を受ける準備を行います。	P24
—	GetEZ	EasyBuilder で設定した結果を受け取ります。	P24
—	ResetEZ	GetEZ 用のフラグを立てます。	P25
—	SetTimeoutNM	In-Sight との通信のタイムアウト時間を設定します。	P25
—	GetTimeoutNM	In-Sight との通信のタイムアウト時間を取得します。	P26
—	ExecuteCommand	生データの送信をします。	P26
—	RecievePacket	生データを受信します。	P26

2.6. 基本通信

2.6.1. CaoController::Execute(“NativeMode”) コマンド

ネイティブモードコマンドの送受信を行います。

書式 NativeMode (<bstrSyntax>, [<bstrOption>])

<bstrSyntax> : [in] シンタックス (VT_BSTR)

<bstrOption> : [in] BSTR オプション(VT_BOOL)

False	戻り値を VT_UI1 VT_ARRAY で返します. (デフォルト)
True	戻り値を VT_BSTRで返します.

戻り値 : [out] ステータスコードおよびコマンドの戻り値 (VT_UI1 | VT_ARRAY or VT_BSTR)

In-Sight とネイティブモードで通信します. bstrSyntax にはコマンド名とパラメータを文字列で設定します。

使用例

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("NativeMode", Array("GF", "BSTR=True"))

vntResult : "1
MyJob1.job"
```

2.7. ファイルコマンドおよびジョブコマンド

2.7.1. CaoController::Execute(“LoadFile”) コマンド

指定したジョブをメモリからロードし、アクティブなジョブにします。

注意: In-Sight デバイスはオフラインになっている必要があります。

書式 LoadFile (<bstrFileName>)

<bstrFileName> : [in] ジョブのファイル名(VT_BSTR)

戻り値 : なし

In-Sight プロセッサのメモリから, bstrFileName で指定したジョブをロードし、アクティブなジョブにします。

使用例

```
caoCtrl.Execute "LoadFile", "MyJob1.job"
```

2.7.2. CaoController::Execute(“StoreFile”) コマンド

In-Sight プロセッサのメモリに現在のジョブを保存します。

注意: In-Sight デバイスはオフラインになっている必要があります。

書式 StoreFile (<bstrFileName>)

<bstrFileName> : [in] ジョブのファイル名(VT_BSTR)

戻り値 : なし

In-Sight プロセッサのメモリに現在のジョブ bstrFileName の名前をつけて保存します。

使用例

```
caoCtrl.Execute "StoreFile", "MyJob1.job"
```

2.7.3. CaoController::Execute ("DeleteFile") コマンド

In-Sight プロセッサのメモリ内の指定したジョブスロットからジョブを削除します。

注意: In-Sight デバイスはオフラインになっている必要があります。

書式 DeleteFile (<bstrFileName >)

<bstrFileName > : [in] ジョブのファイル名(VT_BSTR)

戻り値 : なし

bstrFileName で指定したジョブを削除します。

使用例

```
caoCtrl.Execute "DeleteFile", "MyJob1.job"
```

2.7.4. CaoController::Execute ("GetFile") コマンド

In-Sight デバイス上のアクティブなジョブのファイル名を返します。

注意: このコマンドを正常に実行するために、アクティブなジョブは保存しておく必要があります。ジョブがドラッグアンドドロップされたものである場合は、そのジョブのファイル名が返されます。

書式 GetFile ()

引数 : なし

戻り値 : [out] bstrFilename(VT_BSTR)

アクティブなジョブのファイル名を VT_BSTR で取得します

使用例

```
Dim bstrFilename as String  
bstrFilename = caoCtrl.Execute("GetFile")  
  
bstrFilename : "MyJob1.job"
```

2.7.5. CaoController::Execute (“SetJob”) コマンド

In-Sight プロセッサのメモリ内のジョブスロットからジョブをロードし、それをアクティブなジョブとします。

注意: ジョブ ID 番号の機能を使用するには、ロードするジョブが数値 0 ~ 999 の接頭辞を付けて保存されている必要があります。非ネットワーク対応 In-Sight センサーとの下位互換性のためには、接頭辞は 0 ~ 19 の範囲である必要があります。

In-Sight システムはオフラインになっている必要があります。

書式

SetJob (<IID>)

<IID> : [in] ジョブの ID(VT_I4) (0~999)

戻り値 : なし

IID で指定したジョブをアクティブなジョブにします。

使用例

```
caoCtrl.Execute "SetJob ", 2
```

```
アクティブなジョブ : "2MyJob1.job"
```

2.7.6. CaoController::Execute (“StoreJob”) コマンド

In-Sight プロセッサ上のメモリ内の、指定したジョブスロットに現在のジョブを保存します。

書式

StoreJob (<IID>, <bstrJobName>)

<IID> : [in] ジョブの ID(VT_I4) (0~19)

<bstrJobName> : [in] ジョブの名前(VT_BSTR)

戻り値 : なし

現在のジョブを bstrJobName で名前をつけて、IIDのスロットに保存します。

使用例

```
caoCtrl.Execute "StoreJob", Array(3, "TEST.job")
```

2.7.7. CaoController::Execute (“DeleteJob”) コマンド

In-Sight プロセッサのメモリ内の指定したジョブスロットからジョブを削除します。

注意: 下位互換性のために、ジョブの ID 番号は 0 ~ 19 の範囲である必要があります。

In-Sight デバイスはオフラインになっている必要があります。

書式

DeleteJob (<IID>)

<IID> : [in] ジョブの ID(VT_I4) (0~999)

戻り値 : なし

IID で指定したジョブを削除します。

使用例

```
caoCtrl.Execute "DeleteJob", 2
```

2.7.8. CaoController::Execute ("GetJob") コマンド

In-Sight デバイス上のアクティブなジョブの ID を返します。

注意: ジョブ ID 番号の機能を使用するには、ロードするジョブが数値 0 ~ 999 の接頭辞を付けて保存されている必要があります。非ネットワーク対応 In-Sight センサーとの下位互換性に対しては、接頭辞が 0 ~ 19 である必要があります。

このコマンドを正常に実行するために、アクティブなジョブは、数値の接頭辞を付けて保存しておく必要があります。ジョブがドラッグアンドドロップされたものである場合、ファイル名に数値の接頭辞が付いていなければ、このコマンドは正常に実行されません。

書式

GetJob ()

引数 : なし

戻り値 : [out] ジョブの ID(VT_I4) (0~999)

VT_I4 でアクティブなジョブ ID を返します。

使用例

```
Dim IID as Long
IID = caoCtrl.Execute("GetJob")
```

```
IID : 2
```

2.8. 設定コマンドおよびセル値コマンド

2.8.1. CaoController::Execute ("GetValue") コマンド

指定したセルに含まれている値を返します。

注意: In-Sight のセルに数値が含まれている場合、その数値が整数か浮動小数点数値かにかかわらず、GetValue コマンドで要求すると、小数第 3 位までにフォーマットされた浮動小数点数値が返されます。構造体のような印刷されない文字がセルに含まれている場合、印刷されない文字の代わりにシャープ記号 (#) が送信されます。セルが空白の場合は、ヌル文字が送信されます。

書式

GetValue (<bstrRow>, <lCol>)

<lCol> : [in] 取得する列文字(VT_BSTR) (A~Z)

<bstrRow> : [in] 取得する行番号(VT_I4) (0~399)

戻り値 : [out] セルの値(VT_BSTR)

<bstrRow>と<lCol>で指定したセルの値を VT_BSTR で返します。

使用例

```
Dim bstrVal as String
bstrVal = caoCtrl.Execute("GetValue", Array("B", 1))

bstrVal: 1.000
```

2.8.2. CaoController::Execute ("SetInteger") コマンド

セル内に格納されているコントロールを、指定した整数値に設定します。コントロールは、EditInt, Checkbox, または ListBox タイプである必要があります。

書式

SetInteger(<bstrRow>, <lCol>, <lValue>)

<lCol> : [in] 設定する列文字(VT_BSTR) (A~Z)
<bstrRow> : [in] 設定する行番号(VT_I4) (0~399)
<lValue> : [in] 設定する値(VT_I4)
戻り値 : なし

<bstrRow>と<lCol>で指定したセルの値を<lValue>の値に設定します。

使用例

```
caoCtrl.Execute "SetInteger", Array("B", 1, 100)
```

2.8.3. CaoController::Execute ("SetFloat") コマンド

セル内に格納されている編集ボックスコントロールを、指定した浮動小数点値に設定します。編集ボックスコントロールは EditFloat タイプである必要があります。

書式

SetFloat (<bstrRow>, <lCol>, <fValue>)

<lCol> : [in] 設定する列文字(VT_BSTR) (A~Z)
<bstrRow> : [in] 設定する行番号(VT_I4) (0~399)
<fValue> : [in] 設定する値(VT_R4)
戻り値 : なし

<bstrRow>と<lCol>で指定したセルの値を<fValue>の値に設定します。

使用例

```
caoCtrl.Execute "SetFloat", Array("B", 1, 10.130)
```

2.8.4. CaoController::Execute ("SetString") コマンド

セル内に格納されている編集ボックスコントロールを、指定した文字列に設定します。編集ボックスは

EditString タイプである必要があります。

書式 SetString (<bstrRow>, <lCol>, <bstrValue>)

<lCol> : [in] 設定する列文字(VT_BSTR) (A~Z)

<bstrRow> : [in] 設定する行番号(VT_I4) (0~399)

<bstrValue> : [in] 設定する値(VT_BSTR)

戻り値 : なし

<bstrRow>と<lCol>で指定したセルの値を<bstrValue>の値に設定します。

使用例

```
caoCtrl.Execute "SetString", Array("B", 1, "Test")
```

2.8.5. CaoController::Execute ("GetInfo") コマンド

In-Sight デバイスのシステム情報を返します。

書式 GetInfo ()

引数 : なし

戻り値 : [out] システム情報(VT_BSTR | VT_ARRAY)

In-Sight デバイスのシステム情報を取得します。結果はシリアル番号, アプリケーションバージョン, モニタバージョン, MAC アドレス, ビルドの日付の順番に格納されています。

使用例

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

vntResult : "Serial Number:        Z83560368",
           "Application Version: 4.05.00 (233)",
           "Monitor Version:     4.01",
           "MAC Address:         00-d0-24-02-41-a1",
           "Date of Build:       Apr 4 2011, 13:18:09"
```

2.8.6. CaoController::Execute ("StoreSettings") コマンド

In-Sight システムの設定を proc.set ファイルに保存します。

書式 StoreSettings ()

引数 : なし

戻り値 : なし

システム設定を保存します。

使用例

```
caoCtrl.Execute "StoreSettings"
```

2.8.7. CaoController::Execute (“SetIPLock”) コマンド

In-Sight デバイスの IP アドレスが無断で変更されることを防ぎます。「アクセス不可」が設定されている場合、アクセス権が「プロテクト」または「アクセス不可」のユーザが [カメラ接続マネージャ] で行った IP アドレスの変更は、保存されません。

注意: IP アドレスがアクセス不可の場合、アクセス権が「プロテクト」または「アクセス不可」のユーザは、In-Sight デバイスにログオンできますが、IP アドレスを変更することはできません。

書式 StoreSettings (<lNum>)

<lNum> : [in] 設定するパラメータ(VT_I4)
0 : IP アドレスをアクセス可能にします。
1 : IP アドレスをアクセス不可にします。

戻り値 : なし

IP アドレスのロック状態を<lNum>で指定した状態にします。

使用例

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

2.8.8. CaoController::Execute (“GetIPLock”) コマンド

In-Sight センサー上での IP アドレスのセキュリティ状態を返します。SetIPLock コマンドを使用すると、IP アドレスが無断で変更されることを防ぐ機能を確認できます。

書式 GetIPLock ()

引数 : なし

戻り値 : [out] IP アドレスのロック状態(VT_I4)
0 : IP アドレスアクセス可能。
1 : IP アドレスアクセス不可。

IP アドレスのロック状態を VT_I4 で返します。

使用例

```
Dim lLock as long  
lLock = caoCtrl.Execute (“GetIPLock”)  
lLock : 1
```

2.9. 実行コマンドおよびオンラインコマンド

2.9.1. CaoController::Execute (“SetOnline”) コマンド

In-Sight デバイスをオンラインまたはオフラインに設定します。

注意: 以前に ([システム] ツールバーから) 手動で、またはディスクリート入力信号で、In-Sight デバイスをオフラインに設定した場合、このコマンドを使用して In-Sight デバイスをオンラインにすることはできません。

書式

SetOnline (<INum>)

<INum> : [in] In-Sight デバイスのオンライン状態 (VT_I4)
0 : In-Sight デバイスをオフラインに設定します。
1 : In-Sight デバイスをオンラインに設定します。

戻り値 : なし

In-Sight デバイスのオンライン状態を INum で設定します。

使用例

```
caoCtrl.Execute "SetOnline", 1
```

2.9.2. CaoController::Execute (“GetOnline”) コマンド

In-Sight プロセッサのオンライン状態を返します。

書式

GetOnline ()

引数 : なし

戻り値 : [out] In-Sight デバイスのオンライン状態 (VT_I4)
0 : In-Sight プロセッサは現在オフラインです。
1 : In-Sight プロセッサは現在オンラインです。

In-Sight プロセッサのオンライン状態を INum で返します。

使用例

```
Dim lState as Long  
lState = caoCtrl.Execute("GetOnline")  
  
lState : 1
```

2.9.3. CaoController::Execute (“SetEvent”) コマンド

指定したイベントにトリガをかけます。

注意: In-Sight デバイスはオンラインになっている必要があります。

書式

SetEvent (<IEventNum>)

<IEventNum> : [in] 設定するイベントコード (VT_I4)

0～7：ソフトトリガを指定します。

8：画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。

戻り値 : なし

In-Sight にソフトウェアトリガをかけます。

使用例

```
caoCtrl.Execute "SetEvent", 8
```

2.9.4. CaoController::Execute ("SetEventAndWait") コマンド

指定したイベントにトリガをかけ、コマンドが完了してからレスポンスを返します。

注意: In-Sight デバイスはオンラインになっている必要があります。

Timeout 設定時間が短い場合、コマンドが完了する前に E_TIMEOUT が返る場合があります。

書式

SetEventAndWait (<IEventNum>)

<IEventNum> : [in] 設定するイベントコード (VT_I4)

0～7：ソフトトリガを指定します。

8：画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。

注意: カメラトリガは、In-Sight 1700 シリーズ、In-Sight 1000 シリーズ、または In-Sight 4000 シリーズのセンサーではサポートされていません。

戻り値 : なし

In-Sight にソフトウェアトリガをかけます。

使用例

```
caoCtrl.Execute "SetEventAndWait", 8
```

2.9.5. CaoController::Execute ("SendMessage") コマンド

ASCII 文字列をネイティブモード接続経由で In-Sight のスプレッドシートに送信します。スプレッドシートのイベントにトリガをかけることもできます。

書式

SendMessage (<bstrMessage>, [<INum>])

<bstrMessage> : [in] 設定する文字列 (VT_BSTR)

注意: In-Sight のコマンドでは文字列は引用符で囲む必要があ

りますが、本コマンドでは囲む必要がありません。

<INum>

[in] 設定するイベントコード (VT_I4)

0~7: ソフトトリガを指定します。

8: 画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。

注意: これはオプションのパラメータです。

戻り値 : なし

In-Sight に bstrMessage を送信します。INum を指定する事でイベントトリガをかけることもできます。

使用例

```
caoCtrl.Execute "SendMessage", Array("hoge", 8)
```

2.10. ネイティブモードの拡張コマンド

2.10.1. CaoController::Execute ("GetFilelist") コマンド

In-Sight デバイスのメモリに格納されているファイル名のリストを返します。

注意: In-Sight のコマンドではファイル数を返しますが、本コマンドでは返しません。

書式

GetFilelist ()

引数 : なし

戻り値 : [out] ファイルリスト (VT_BSTR | VT_ARRAY)

In-Sight に保存されているファイルを VT_BSTR|VT_ARRAY で返します。

使用例

```
bstrFilenames = caoCtrl.Execute("GetFilelist")
```

```
bstrFilenames : array("TEST.job", "hoge.job")
```

2.11. 独自拡張コマンド

2.11.1. CaoController::Execute ("SendMessageAndWait") コマンド

SendMessage コマンドを発行した後に、In-Sight の WriteMessage 関数によって出力される文字列を受信します。WriteMessage 関数で送られた文字列に終端記号が設定されていない場合は、タイムアウト時間まで待機し、受信できた分だけ返します。終端記号は In-Sight の FormatString 関数等で設定してください。

注意: In-Sight デバイスはオンラインになっている必要があります。

書式

SendMessageAndWait ([<bstrMessage>], [<INum>], [<ITerminateNum>])

- <bstrMessage> : [in] 設定する文字列 (VT_BSTR)
 注意: In-Sight のコマンドでは文字列は引用符で囲む必要がありますが、本コマンドでは囲む必要がありません。省略時は空文字が送られます。
- <lNum> [in] 設定するイベントコード (VT_I4)
 0~7: ソフトトリガを指定します。
 8: 画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。
 注意: 省略時は 8 が使用されます。
- <lTerminateNum> [in] 設定する終端記号番号 (VT_I4)
 0: 無し
 1: 復帰(CR)
 2: 改行(FL)
 3: CR+LF
 注意: 省略時は 1 が使用されます。
- 戻り値 : [out] 出力されたテキスト文字列 (VT_BSTR)

In-Sight にイベントを送った後に、WriteMessage 関数によって出力される文字列を受信します。WriteMessage 関数が設定されていない場合は、E_TIMEOUT が発生します。

使用例

```
bstrMessage = caoCtrl.Execute("SendMessageAndWait", Array("", 8, 1))
bstrFilenames : "SendMessage"
```

2.11.2. CaoController::Execute ("GetMessage") コマンド

In-Sight の WriteMessage 関数によって出力される文字列を受信します。WriteMessage 関数で送られた文字列に終端記号が設定されていない場合は、タイムアウト時間まで待機し、受信できた分だけ返します。終端記号は In-Sight の FormatString 関数等で設定してください。

注意: In-Sight デバイスはオンラインになっている必要があります。

書式

- GetMessage([<lTerminateNum>])
- <lTerminateNum> [in] 設定する終端記号番号 (VT_I4)
 0: 無し
 1: 復帰(CR)
 2: 改行(FL)
 3: CR+LF

注意: 省略時は 1 が使用されます。

戻り値 : [out] 出力されたテキスト文字列 (VT_BSTR)

WriteMessage 関数によって出力される文字列を受信します。WriteMessage 関数が設定されていない場合は、E_TIMEOUT のエラーが発生します。

使用例

```

bstrMessage = caoCtrl.Execute("GetMessage")
bstrFilenames : "SendMessage"

```

2.11.3. CaoController::Execute ("SendMessageAndGetEZ") コマンド

SendMessage コマンドを発行した後に、In-Sight の EasyBuilder の通信 (TCP/IP) によって出力される文字列を受信します。EasyBuilder の設定は 3.2 章を参照してください。

注意: In-Sight デバイスはオンラインになっている必要があります。

書式

SendMessageAndGetEZ ([<bstrMessage>], [<lNum>], [<lTimeOut>])

<bstrMessage> : [in] 設定する文字列 (VT_BSTR)

注意: In-Sight のコマンドでは文字列は引用符で囲む必要がありますが、本コマンドでは囲む必要がありません。省略時は空文字が送られます。

<lNum> [in] 設定するイベントコード (VT_I4)

0~7: ソフトトリガを指定します。

8: 画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。

注意: 省略時は 8 が使用されます。

<lTimeOut> [in] 受信待ちする最大時間(msec)(VT_UI4)

注意: 省略時は 500msec 待ちます。

戻り値 : [out] 出力されたテキスト文字列 (VT_BSTR)

In-Sight にイベントを送った後に、EasyBuilder の通信設定で設定された文字列を受信します。通信設定がされていない場合や受信待ち時間を超えた場合は、E_TIMEOUT のエラーが発生します。AddController で EZPort オプションが指定されていない場合は E_EZERROR のエラーが発生します。

使用例

```

bstrMessage = caoCtrl.Execute("SendMessageAndGetEZ", Array("", 8, 1000))
bstrFilenames : "SendMessage"

```

2.11.4. CaoController::Execute (“SendMessageEZ”) コマンド

SendMessage コマンドを発行した後に、In-Sight の EasyBuilder の通信 (TCP/IP) によって出力される文字列を受信する準備をします。文字列の受信は GetEZ コマンドを使用してください。

注意: In-Sight デバイスはオンラインになっている必要があります。

書式 SendMessageEZ ([<bstrMessage>], [<lNum>])

<bstrMessage> : [in] 設定する文字列 (VT_BSTR)

注意: In-Sight のコマンドでは文字列は引用符で囲む必要がありますが、本コマンドでは囲む必要がありません。省略時は空文字が送られます。

<lNum> [in] 設定するイベントコード (VT_I4)

0~7: ソフトトリガを指定します。

8: 画像を取り込み、スプレッドシートを更新します。このオプションでは、AcquireImage 関数のトリガ引数を、カメラ、外部トリガ、または手動トリガに設定する必要があります。

注意: 省略時は 8 が使用されます。

戻り値 : なし

GetMessage コマンドで受信できる結果は直前に発行された SendMessageEZ コマンドで送信された結果だけです。

使用例

```
Call caoCtrl.Execute("SendMessageEZ", Array("", 8))
```

2.11.5. CaoController::Execute (“GetEZ”) コマンド

In-Sight の EasyBuilder の通信 (TCP/IP) によって出力される文字列を受信します。GetEZ コマンドで結果を受信する際には、SendMessageEZ コマンドでトリガをかけるか、ResetEZ コマンドを利用してください。

注意: In-Sight デバイスはオンラインになっている必要があります。AddController で EZPort オプションが指定されていない場合は E_EZERROR のエラーが発生します。

書式 GetEZ ([<lTimeOut >])

<lTimeOut > : [in] 受信待ちする最大時間(msec)(VT_UI4)

注意: 省略時は 500msec 待ちます。

戻り値 : [out] 出力されたテキスト文字列 (VT_BSTR)

GetEZ コマンドで結果を受信できるのは、SendMessageEZ コマンドでトリガを発生させた時か ResetEZ コマ

ンドで受信フラグを立てた時だけです。上記以外の方法でトリガが発生させた場合や EasyBuilder で結果出力が設定されていない場合には、E_TIMEOUT のエラーが発生します。

使用例

```
Call caoCtrl.Execute("SendMessageEZ", Array("", 8))
bstrResult = caoCtrl.Execute("GetEZ", 1000)
```

2.11.6. CaoController::Execute ("ResetEZ") コマンド

GetEZ コマンドで結果を受信するためのフラグを設定します。このコマンドを発行すると、1 度だけ EasyBuilder で設定した結果を GetEZ コマンドで取得することができます。

書式

ResetEZ

引数 : 無し

戻り値 : 無し

ResetEZ コマンドを発行後に受信した結果を GetEZ コマンドで取得することができます。GetEZ コマンドで結果を受信したい場合には、結果が送られてくる前に ResetEZ コマンドを使用してください。

使用例

```
Call caoCtrl.Execute("ResetEZ")
bstrResult = caoCtrl.Execute("GetEZ", 1000)
```

2.11.7. CaoController::Execute ("SetTimeoutNM") コマンド

ネイティブモードで通信する際のタイムアウト時間を変更します。初期値は AddController のオプション (Timeout) で設定した値になっています。

書式

SetTimeoutNM (<ITimeOut >)

<ITimeOut > : [in] 設定するタイムアウト時間(msec)(VT_UI4)

戻り値 : なし

使用例

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

2.11.8. CaoController::Execute (“GetTimeoutNM”) コマンド

ネイティブモードで通信する際のタイムアウト時間を取得します。

書式

GetTimeoutNM ()

引数 : なし

戻り値 : [out] 設定されたタイムアウト時間(msec)(VT_UI4)

使用例

```
ITimeout = caoCtrl.Execute("GetTimeoutNM")
```

2.11.9. CaoController::Execute (“ExecuteCommand”) コマンド

生データを送信します。

書式

ExecuteCommand (< Raw Data >)

< Raw Data > : [in] 送信する生データ(VT_BSTR)

戻り値 : なし

使用例

```
Call caoCtrl.Execute("ExecuteCommand ", "GetTimeoutNM")
```

2.11.10. CaoController::Execute (“RecievePacket”) コマンド

生データを受信します。

書式

RecievePacket ()

引数 : なし

戻り値 : [out] 受信した生データ(VT_BSTR)

使用例

```
strRowReciveData = caoCtrl.Execute("RecievePacket")
```

3. 画像処理結果の取得方法

In-Sight から結果を取得する方法は大きく分けて2つあります。

1. スプレッドシートのデータをネイティブモード経由で取得する
2. EasyBuilder で通信設定を行って、結果を取得する

3.1. ネイティブモード経由での取得

ネイティブモード経由では WriteMessage 関数で出力された文字列を取得する方法とスプレッドシートの値を取得する方法があります。

3.1.1. WriteMessage 関数の利用

WriteMessage 関数で出力された文字列を受け取るには、FormatString 関数で出力したい結果を文字列として作成します。FormatString 関数で設定する終端記号は SendMessageAndWait コマンドで結果を受信する際に使用します。FormatString 関数で作成した文字列を WriteMessage 関数で指定して出力してください。

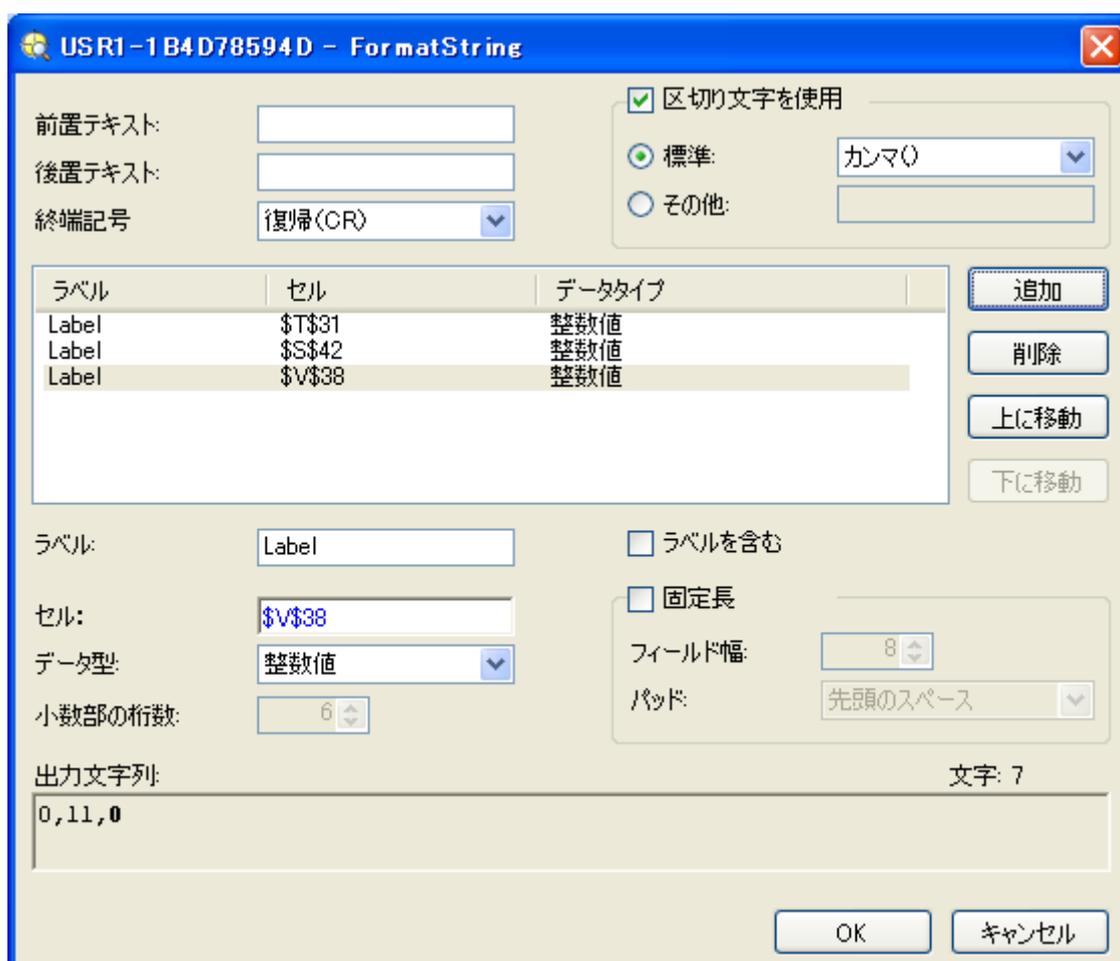


図 3-1 FormatString 関数



図 3-2 WriteMessage 関数

使用例

```
Private Sub Form_Load()  
    Dim caoEng As CaoEngine  
    Dim caoCtrl As CaoController  
  
    Set caoEng = New CaoEngine  
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",  
"Conn=eth:127.0.0.1, Timeout=1000")  
  
    Dim strResult As String  
  
    strResult = caoCtrl.Execute("SendMessageAndWait", Array("", 8, 1))  
End Sub
```

3.1.2. スプレッドシートの値の取得

スプレッドシートの値を取得するには、GetValue コマンドを利用するか AddVariable メソッドでセルの値を参照する変数を作成します。

使用例

```
Private Sub Form_Load()  
    Dim caoEng As CaoEngine  
    Dim caoCtrl As CaoController  
  
    Set caoEng = New CaoEngine  
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",  
    "Conn=eth:127.0.0.1, Timeout=1000")  
  
    Dim strResult As String  
  
    Call caoCtrl.Execute("SendMessage", Array("", 8))  
    strResult = caoCtrl.Execute("GetValue", Array("0", 60))  
  
End Sub
```

3.2. EasyBuilder での結果取得

EasyBuilder で作成した画像処理結果を取得するためには、EasyBuilder の通信機能を利用します。通信設定で出力したい結果を設定し、SendMessageAndGetEZ コマンドで結果を取得したり CaoController のイベント出力として結果を受信することができます。下記に EasyBuilder で通信設定を行う方法を示します。

3.2.1. EasyBuilder の設定

始めに、図 3-3 の通信ボタンを押して通信デバイスの設定画面を開きます。



図 3-3 通信設定

次に図 3-4 のデバイスの追加ボタンを押して、下記のデバイスを追加してください。

デバイス : その他

プロトコル : TCP/IP

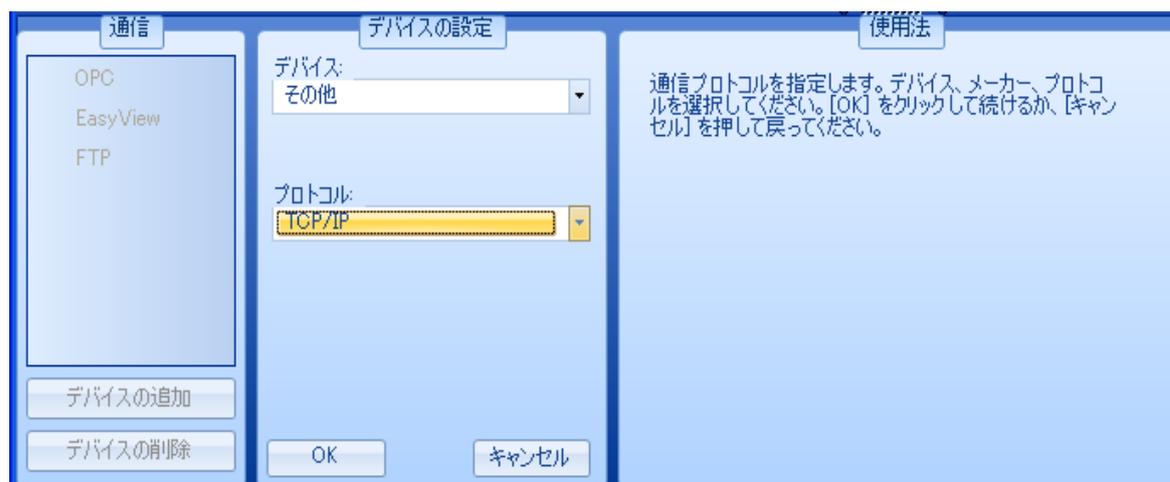


図 3-4 通信デバイスの設定

次に図 3-5 出力先の設定を行います。

サーバホスト名 : プロバイダを使用する PC の IP アドレス

ポート : AddController の EZPort オプションで指定するポート番号

終端記号 : CRLF (固定)



図 3-5 出力先の設定

次にフォーマット出力文字列のタブを押し、カスタムフォーマットボタンを押して出力したい結果を登録します。その際に終端記号はなしにしてください。

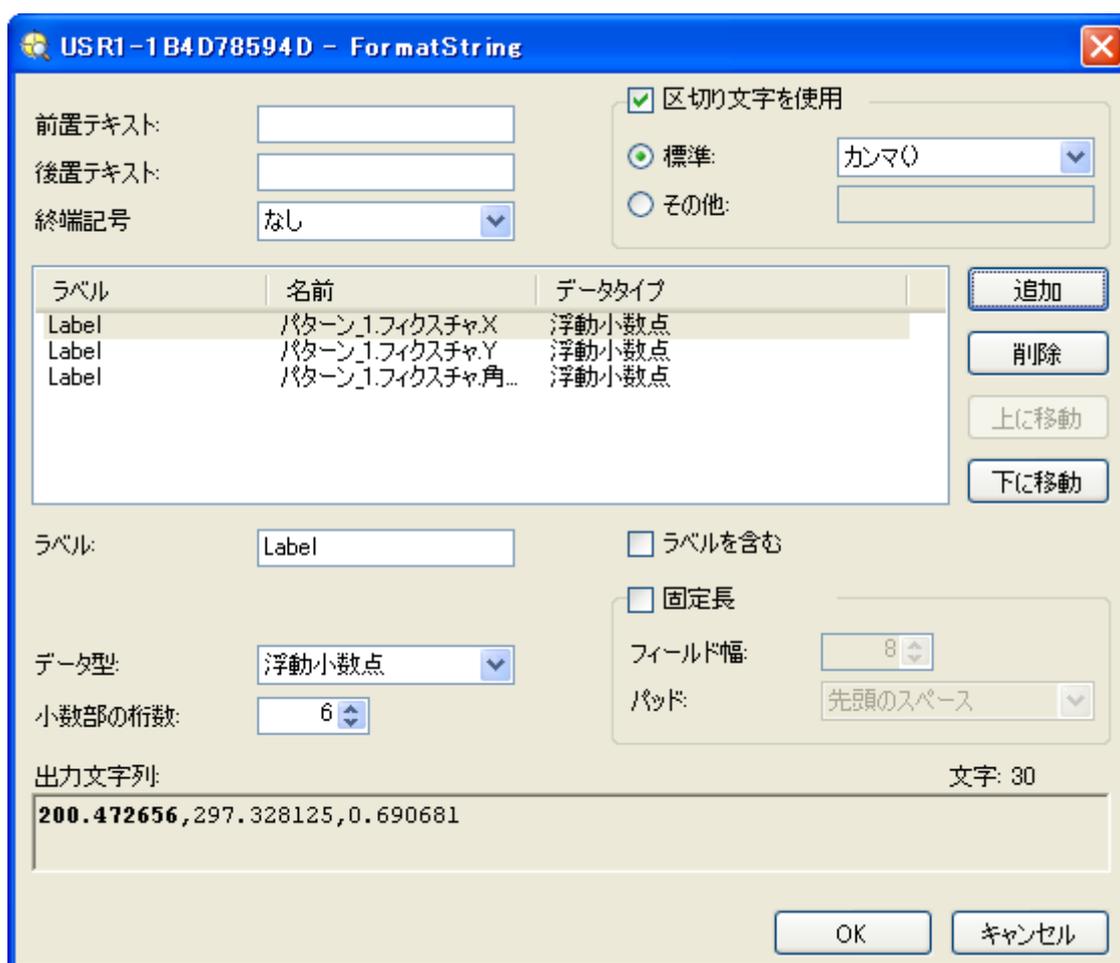


図 3-6 出力結果の登録

以上の設定でオンラインモードにすると、トリガがかかった後に設定した PC へ結果を送信します。次に結果の受信方法を示します。

3.2.2. EasyBuilder の結果取得方法

EasyBuilder で設定した結果を受信するには、AddController のオプションに”EZPort=ポート番号”で EasyBuilder で指定した出力ポートを指定する必要があります。

結果の受信コマンドは”SendMessageAndGetEZ”コマンドか”SendMessageEZ”と”GetEZ”コマンドを使用します。”SendMessageAndGetEZ”コマンドはソフトウェアトリガをかけた後、結果を受信するまで待機します。”SendMessageEZ”はソフトウェアトリガだけかけますので、その後”GetEZ”コマンドで結果を受信してください。

上記コマンドを使ってソフトウェアトリガをかけない場合は、CaoController の OnMessage イベントで結果を受信するか、ResetEZ コマンドで GetEZ コマンドを使用できるようにフラグを立てます。

以下に使用例を示します。

3.2.2.1. SendMessageAndGetEZ の使用例

使用例

```
Private Sub Form_Load()  
    Dim caoEng As CaoEngine  
    Dim caoCtrl As CaoController  
  
    Set caoEng = New CaoEngine  
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",  
"Conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")  
  
    Dim strResult As String  
    strResult = caoCtrl.Execute("SendMessageAndGetEZ", Array("", 8, 1000))  
  
End Sub
```

3.2.2.2. SendMessageEZ と GetEZ の使用例

使用例

```
Private Sub Form_Load()  
    Dim caoEng As CaoEngine  
    Dim caoCtrl As CaoController  
  
    Set caoEng = New CaoEngine  
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",  
"Conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")  
  
    Dim strResult As String  
    Call caoCtrl.Execute("SendMessageEZ", Array("", 8))  
  
    ' 他の処理  
  
    strResult = caoCtrl.Execute("GetEZ", 1000)  
End Sub
```

3.2.2.3. ResetEZ と GetEZ の使用例

使用例

```
Private Sub Form_Load()  
    Dim caoEng As CaoEngine  
    Dim caoCtrl As CaoController  
  
    Set caoEng = New CaoEngine  
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",  
"Conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")  
  
    Dim strResult As String  
    Call caoCtrl.Execute("ResetEZ")  
  
    ' トリガ発生  
    Call caoCtrl.Execute("SetEvent", 8)  
  
    strResult = caoCtrl.Execute("GetEZ", 1000)
```

End Sub

3.2.2.4. イベントでの使用例

使用例

```
Dim caoEng As CaoEngine
Dim WithEvents caoCtrl As CaoController

Private Sub Form_Load()
    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"Conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")
End Sub

Private Sub Command1_Click()
    Call caoCtrl.Execute("SetEvent", 8)
End Sub

Private Sub caoCtrl_OnMessage(ByVal pICaoMess As CAOLib.ICaoMessage)
    Dim strResult As String

    Select Case pICaoMess.Number
    Case 1
        strResult = pICaoMess.Value
        Debug.Print strResult
    End Select
End Sub
```
