

# SIEMENS PLCSIM プロバイダ

Version 1.1.0

ユーザーズ ガイド

September 5, 2024

備考：

**【改版履歴】**

バージョン	日付	内容
1.0.0	2016-09-05	初版.
	2020-11-16	ライセンス追加事項追記.
1.1.0	2024-09-05	適切に文字列の解放処理が行われていなかった箇所を修正.

**【対応機器】**

機種	バージョン	注意事項
S7-PLCSIM Advanced	V1.0	

**【ご注意】**

**本プロバイダを使用する場合は別途“SIEMENS PLCSIM プロバイダ”ライセンスが必要です.**

## 目次

1. はじめに .....	4
1.1. ライセンスの追加 .....	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
2.2.2. CaoController::AddVariable メソッド .....	8
2.2.3. CaoController::get_VariableNames メソッド .....	8
2.2.4. CaoController::Execute メソッド .....	8
3. 変数リファレンス .....	9
3.1. インスタンスリスト取得モード .....	9
3.2. インスタンス制御モード .....	10
4. コマンドリファレンス .....	20
4.1. インスタンスリスト取得モード .....	20
4.2. インスタンス制御モード .....	21
5. エラーコード .....	23
6. サンプルプログラム .....	24

## 1. はじめに

本書は、SIEMENS 製の PLC シミュレーション: S7-PLC Simulation Advanced (シミュレーションソフトと記述) を制御するためのプロバイダのユーザーズマニュアルです。シミュレーションソフトは 1 ホストにつき最大 1 つ起動することができ、シミュレーションソフト 1 つにつき複数の仮想コントローラを定義 (インスタンスと呼びます) できます。

## 1.1. ライセンスの追加

本プロバイダを使用可能にするには ORiN2 SDK をインストール後、別途「SIEMENS PLCSIM プロバイダ」ライセンスを入力する必要があります。評価用にインストールする場合は下記のライセンスキーをご使用ください。

**SIEK-R15W-USE5-ETJB** （評価用 3 ヶ月）

下記に「SIEMENS PLCSIM プロバイダ」ライセンスの追加手順を示します。

1. CaoConfig を起動し、[Cao Provider] タブを選択する
2. Provider List から [SIEMENS PLCSIM CAO Provider] 項目を選択する
3. License 項目の [...] ボタンをクリックする
4. ORiN2 License Manager で [Add] ボタンをクリックする
5. 入手したライセンスキーを入力後、[OK] ボタンをクリックする
6. [Close] ボタンをクリックし、CaoConfig を終了する

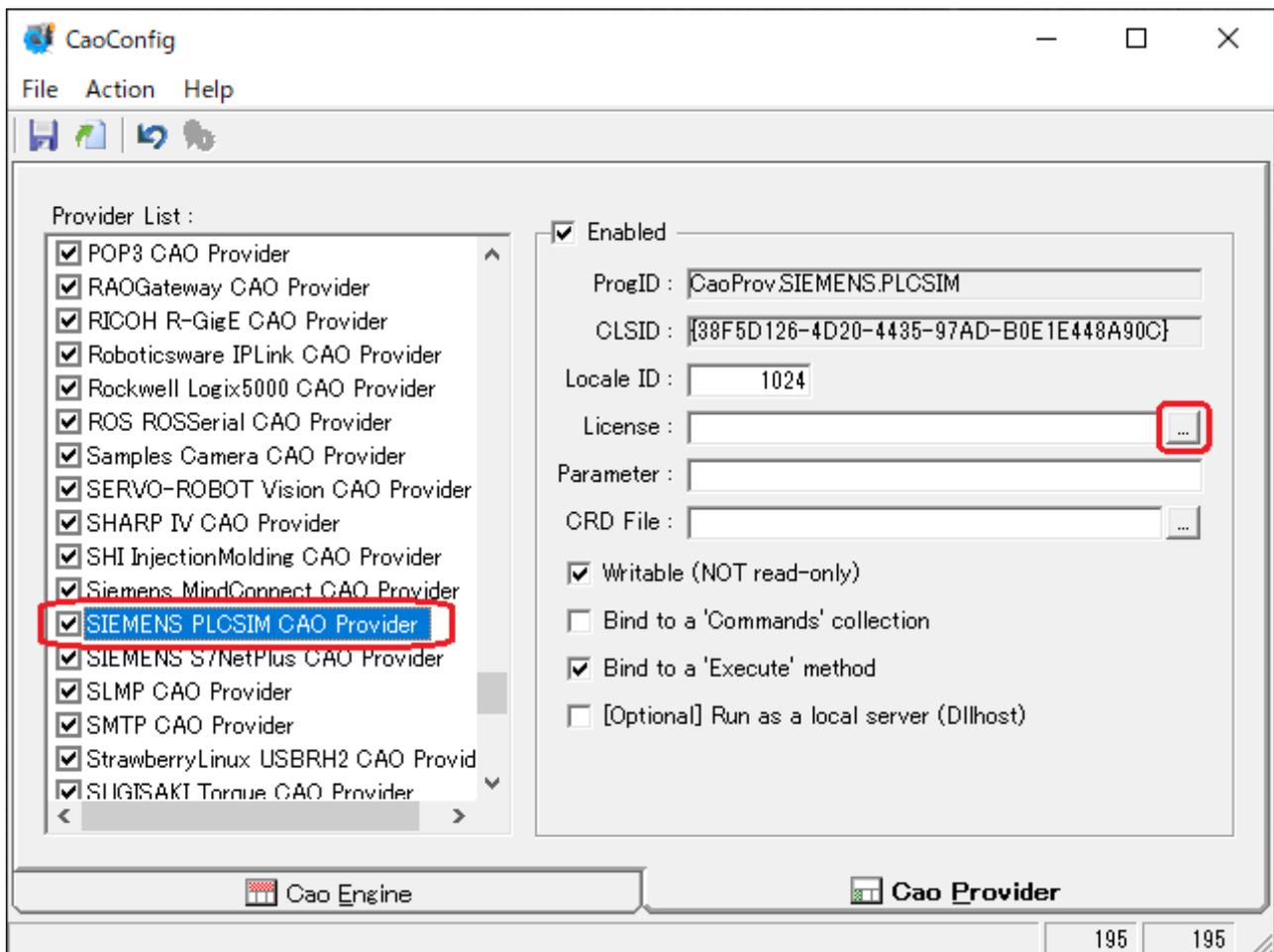


図 1-1 「SIEMENS PLCSIM プロバイダ」ライセンス追加

## 2. プロバイダの概要

### 2.1. 概要

SIEMENS PLCSIM プロバイダは DLL (Dynamic Link Library) となっており、その詳細は表 2-1 のようになっています。

表 2-1 SIEMENS PLCSIM プロバイダ

ファイル名	CaoProvSIEMENSPLCSIM.dll
ProgID	GaoProv.SIEMENS.PLCSIM
レジストリ登録	regsvr32 CaoProvSIEMENSPLCSIM.dll
レジストリ登録の抹消	regsvr32 /u CaoProvSIEMENSPLCSIM.dll

本プロバイダには大きく分けて 2 つの動作モードがあります。

#### 1. インスタンスリスト取得モード

指定されたホストで起動されているシミュレーションソフトに存在するインスタンスリストを取得するモードです。

#### 2. インスタンス制御モード

指定されたインスタンスを制御するモードです。

## 2.2. メソッド・プロパティ

### 2.2.1. CaoWorkSpace::AddController メソッド

Controller オブジェクトの生成時に前述した動作モードとそれぞれの動作モードに必要なオプションを指定します。

以下に、AddController の仕様を示します。

#### 書式

#### AddController

```
(
    "<コントローラ名>",           // コントローラ名(任意)
    "GaoProv.SIEMENS.PLCSIM",     // プロバイダ名(固定)
    "<マシン名>",                 // プロバイダ実行マシン名(未使用)
    "<オプション>"               // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します。

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

オプション	必須	説明	値範囲	デフォルト値
@INSTANCES	--	インスタンス取得モードか否かを指定します。本オプションを指定した場合は、ID、Name オプションは無視されます。 本オプションは名前だけを指定した場合、True と判断します。	True / False	True
Conn=<ETH オプション>	--	接続先のホストを指定します。省略した場合は、ローカルホストのシミュレーションソフトにアクセスします。	----	Dest IP: 必須 Dest Port: 8192 Src IP: 255.255.255.255 Src Port: 0
ID=<インスタンス ID>	--	制御対象インスタンスの ID を指定します。@Instances を指定された場合は無視されます。	0 -	0
Name=<インスタンス名称>	--	制御対象インスタンスの名前を指定します。 @Instances/Id オプションを指定された場合は無視されます。	----	

## (使用例)

- ローカルホストのインスタンスリストを取得したい場合

**@INSTANCES**

- リモートホストのインスタンスを制御したい場合

**Conn=Eth:10.4.9.24:8192, ID=0**

- リモートホストのインスタンスを制御したい場合

**Conn=Eth:10.4.9.24:8192, Name=P-1500**

### 2.2.2. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。動作モードごとに使用できる変数が異なります。

以下に、AddVariable の仕様を示します。

#### 書式

##### AddVariable

```
(  
    "<変数名>", // 変数名  
    ""         // オプション文字列 (未使用)  
)
```

### 2.2.3. CaoController::get\_VariableNames メソッド

変数名リストを取得します。

#### 書式

##### get\_VariableNames

```
(  
    <pVal:VARIANT*> // [out]実行行結果  
)
```

### 2.2.4. CaoController::Execute メソッド

各コマンドの詳細は 4. コマンドリファレンスを参照してください。

#### 書式

##### Execute

```
(  
    <bstrCommand:BSTR> // [in] コマンド名  
    [, <vntParam:VARIANT>] // [in] パラメータ  
    [, <pVal:VARIANT>] // [out]実行行結果  
)
```

## 3. 変数リファレンス

### 3.1. インスタンスリスト取得モード

表 3-1 インスタンスリスト取得モードで使用できる変数一覧を記述します。なお全ての変数で get 操作のみが可能です。

表 3-1 インスタンスリスト取得モードで使用できる コントローラクラス変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	製造元名を返します。	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します。	○	-
@RUNTIMEVERSION	VT_BSTR	ランタイムバージョンを返します。	○	-
@Instances	VT_VARIANT   VT_ARRAY	シミュレーションソフトに存在するすべてのインスタンスリストを取得します。	○	-
	i VT_VARIANT   VT_ARRAY			
	0 VT_I4	ID を保持します。		
	1 VT_BSTR	インスタンス名を保持します。		

### 3.2. インスタンス制御モード

表 3-2 にインスタンス制御モード時に使用できる変数一覧を記述します。なお**全ての**変数で **get** 操作のみが可能です。

表 3-2 インスタンスモードで使用できるコントローラクラス変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKERNAME	VT_BSTR	製造元名を返します。	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します。	○	-
@OPERATINGSTATE	VT_UI1	仮想コントローラの動作状態を取得します。 1: 電源 OFF 2: 電源 ON 途中 3: CPU-STOP 4: CPU が STOP から RUN に移行中 5: CPU-RUN 6: CPU-フリーズ状態 7: 電源 OFF 途中	○	-
@POWER	VT_UI1	仮想コントローラの CPU 状態を取得/設定します。 0: 電源-OFF 1: 電源-ON	○	○
@RUN	VT_UI1	仮想コントローラの CPU 状態を取得/設定します。 0: CPU-STOP 1: CPU-RUN	○	○
@ID	VT_I4	インスタンス ID を取得します。	○	-
@NAME	VT_BSTR	インスタンス名を取得します。	○	-
@CPUTYPE	VT_I4	仮想コントローラの CPU 種類を取得します。	○	-
@TAGINFOS	VT_VARIANT   VT_ARRAY	タグ情報を読み込みます。 ただし、WString 型、String 型のタグは対象外です。	○	-
	i VT_VARIANT   VT_ARRAY			
	0 VT_I4	ID を保持します。		
	1 VT_BSTR	インスタンス名を保持します。		
@TAGNAMES	VT_BSTR   VT_ARRAY	タグ名称を読み込みます。 ただし、WString 型、String 型のタグは対象外です。	○	-
<a href="#">TAG_&lt;??&gt;</a>	タグの型に従います。	タグの値を取得/設定します。	○	○
<a href="#">I_&lt;??&gt;</a>	指定オプションに従います。	入力エリアの指定されたアドレスが保持する値を取得/設定します。	○	○
<a href="#">M_&lt;??&gt;</a>	指定オプションに従います。	マーカエリアの指定されたアドレスが保持する値を取得/設定します。	○	○
<a href="#">O_&lt;??&gt;</a>	指定オプションに従います。	出力エリアの指定されたアドレスが保持する値を取得/設定します。	○	○
<a href="#">AREASIZE_&lt;??&gt;</a>	VT_UI4	指定されたエリアの容量(Byte)を取得します。 本変数で取得した値から 1 減算した値が I/M/O で指定できる Address オプションの最大値となります。	○	-

**TAG\_<??>**

TAG\_<??>は、インスタンスに存在するタグを制御するための変数です。TAG\_の後に任意の文字列を指定してください。本プロバイダでは使用している API の制限により、WString 型、String 型のタグは対象外となっています。その他の型については、表 3-4 の対応表を参照してください。なお、これらの型が配列になる場合は、

対応する型 | VT\_ARRAY

となります。

表 3-3 TAG\_&lt;??&gt;オプション

オプション	必須	説明	設定できる値範囲	デフォルト値
Name=<タグ名称>	○	値を取得/設定する対象のタグ名を指定します。	文字列	---

タグ名称は、階層を「.」でつなげて指定してください。例えば、DataBlock に存在する Robot 構造体の Position 要素を制御したい場合は、「Name=DataBlock.Robot.Position」となります。

なお、指定できるタグ名称のリストを取得したい場合は@TAGNAMES 変数を、タグ名称だけでなく型情報も取得したい場合は@TAGINFOS 変数をそれぞれ使用してください。

表 3-4 タグの型とプロバイダの取得/設定型の対応表

PLC シミュレーションでの型	プロバイダでの型
AOM_IDENT	VT_UI4
Bool	VT_UI1
Byte	VT_UI1
Char	VT_I1
WChar	VT_I2
SInt	VT_I1
USInt	VT_UI1
Int	VT_I2
UInt	VT_UI2
DInt	VT_I4
UDInt	VT_UI4
Lint	VT_I8
ULInt	VT_UI8
Word	VT_UI2
Dword	VT_UI4
LWord	VT_UI8
Real	VT_R4

PLC シミュレーションでの型	プロバイダでの型
LReal	VT_R8
Date	VT_UI2
Date_And_Time	VT_UI8
S5Time	VT_UI2
Time	VT_I4
LTime	VT_I8
Time_Of_Day	VT_UI4
LTime_Of_Day	VT_UI8
String	未対応
WString	未対応
CONN_ANY	VT_UI2
CONN_OUC	VT_UI2
CONN_PRG	VT_UI2
CONN_R_ID	VT_UI4
CREF	VT_VARIANT   VT_ARRAY
BLOCK_TYPE	0   VT_UI1
CB_NUMBER	1   VT_UI2
OFFSET	2   VT_UI4
NREF	VT_VARIANT   VT_ARRAY
AREA	0   VT_UI1
DB_NUMBER	1   VT_UI2
OFFSET	2   VT_UI4
DB_ANY	VT_UI2
DB_DYN	VT_UI2
DB_WWW	VT_UI2
DTL	VT_VARIANT   VT_ARRAY
YEAR	0   VT_UI2
MONTH	1   VT_UI1
DAY	2   VT_UI1
WEEKDAY	3   VT_UI1
HOUR	4   VT_UI1
MINUTE	5   VT_UI1
SECOND	6   VT_UI1
NANSECOND	7   VT_UI4

PLC シミュレーションでの型	プロバイダでの型	
EVENT_ANY	VT_UI4	
EVENT_ATT	VT_UI4	
HW_ANY	VT_UI2	
HW_DEVICE	VT_UI2	
ErrorStruct	VT_VARIANT   VT_ARRAY	
ERROR_ID	0	VT_UI2
FLAGS	1	VT_UI1
REACTION	2	VT_UI1
CODE_ADDRESS	3	VT_VARIANT   VT_ARRAY
BLOCK_TYPE	0	VT_UI1
CB_NUMBER	1	VT_UI2
OFFSET	2	VT_UI4
MODE	5	VT_UI1
OPERAND_NUMBER	6	VT_UI2
POINTER_NUMBER_LOCATION	7	VT_UI2
SLOT_NUMBER_SCOPE	8	VT_UI2
DATA_ADDRESS	9	VT_VARIANT   VT_ARRAY
AREA	0	VT_UI1
DB_NUMBER	1	VT_UI2
OFFSET	2	VT_UI4
HW_DPMaster	VT_UI2	
HW_DPSlave	VT_UI2	
HW_HSC	VT_UI2	
HW_IEPORT	VT_UI2	
HW_INTERFACE	VT_UI2	
HW_IO	VT_UI2	
HW_IOSYSTEM	VT_UI2	
HW_MODULE	VT_UI2	
HW_PTO	VT_UI2	
HW_PWM	VT_UI2	
HW_SUBMODULE	VT_UI2	
IEC_COUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1

PLC シミュレーションでの型	プロバイダでの型	
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_I2
CV	7	VT_I2
IEC_DCOUNT	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_I4
CV	7	VT_I4
IEC_LCOUNT	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_I8
CV	7	VT_I8
IEC_SCOUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_I1
CV	7	VT_I1

PLC シミュレーションでの型	プロバイダでの型	
IEC_UCOUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_UI2
CV	7	VT_UI2
IEC_UDCOUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_UI4
CV	7	VT_UI4
IEC_ULCOUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1
PV	6	VT_UI8
CV	7	VT_UI8
IEC_USCOUNTER	VT_VARIANT   VT_ARRAY	
CU	0	VT_UI1
CD	1	VT_UI1
R	2	VT_UI1
LD	3	VT_UI1
QU	4	VT_UI1
QD	5	VT_UI1

PLC シミュレーションでの型	プロバイダでの型
PV	6 VT_UI1
CV	7 VT_UI1
IEC_TIMER	VT_VARIANT   VT_ARRAY
PT	0 VT_I4
ET	1 VT_I4
IN	2 VT_UI1
Q	3 VT_UI1
IEC_LTIMER	VT_VARIANT   VT_ARRAY
PT	0 VT_I8
ET	1 VT_I8
IN	2 VT_UI1
Q	3 VT_UI1
LDT	VT_I8
OB_ANY	VT_UI2
OB_ATT	VT_I2
OB_CYCLIC	VT_I2
OB_DELAY	VT_I2
OB_DIAG	VT_I2
OB_HWINT	VT_I2
OB_PCYCLE	VT_I2
OB_STARTUP	VT_I2
OB_TIMEERROR	VT_I2
OB_TOD	VT_I2
PIP	VT_UI2
PORT	VT_UI2
RTM	VT_UI2
Struct	VT_ARRAY   VT_VARIANT

#### 使用例

```
CaoController.AddVariable("TAG_Data_block.Robot", "Name=Data_block.Robot")
```

**I\_<??> / M\_<??> / O\_<??>**

I\_<??> / M\_<??> / O\_<??> は、インスタンスの各エリアの値を制御する変数です。<??>にそれぞれ任意の文字列を指定してください。対応するエリアは下記になります。

- I\_<??>: 入力エリア
- M\_<??>: マーカエリア
- O\_<??>: 出力エリア

表 3-5 I\_&lt;??&gt;, M\_&lt;??&gt;, O\_&lt;??&gt;変数のオプション

オプション	必須	説明	設定できる値範囲	デフォルト値																						
VT=<型>	—	使用する型を指定します。大文字小文字は区別しません。	<table border="1"> <tr><td>I1</td><td>符号付 1 バイト整数</td></tr> <tr><td>I2</td><td>符号付 2 バイト整数</td></tr> <tr><td>I4</td><td>符号付 4 バイト整数</td></tr> <tr><td>I8</td><td>符号付 8 バイト整数</td></tr> <tr><td>UI1</td><td>符号なし 1 バイト整数</td></tr> <tr><td>UI2</td><td>符号なし 2 バイト整数</td></tr> <tr><td>UI4</td><td>符号なし 4 バイト整数</td></tr> <tr><td>UI8</td><td>符号なし 8 バイト整数</td></tr> <tr><td>R4</td><td>浮動小数点</td></tr> <tr><td>R8</td><td>倍精度浮動小数点</td></tr> <tr><td>BSTR</td><td>文字列</td></tr> </table>	I1	符号付 1 バイト整数	I2	符号付 2 バイト整数	I4	符号付 4 バイト整数	I8	符号付 8 バイト整数	UI1	符号なし 1 バイト整数	UI2	符号なし 2 バイト整数	UI4	符号なし 4 バイト整数	UI8	符号なし 8 バイト整数	R4	浮動小数点	R8	倍精度浮動小数点	BSTR	文字列	UI1
I1	符号付 1 バイト整数																									
I2	符号付 2 バイト整数																									
I4	符号付 4 バイト整数																									
I8	符号付 8 バイト整数																									
UI1	符号なし 1 バイト整数																									
UI2	符号なし 2 バイト整数																									
UI4	符号なし 4 バイト整数																									
UI8	符号なし 8 バイト整数																									
R4	浮動小数点																									
R8	倍精度浮動小数点																									
BSTR	文字列																									
Address=<アドレス>	—	読み込み/書き込みを開始するアドレスをバイト単位で指定します。	$0 \leq Address \leq (AreaSize - 1)$	0																						
Elem=<要素数>	—	要素数を指定します。ただし VT オプションに BSTR を指定した場合は最大文字数となります。	$1 \leq Elem \leq \frac{(AreaSize - Address)}{Sizeof(VT)}$	1																						
Array=<配列フラグ>	—	Elem が 1 の時に配列型として値を取得するかを指定します。Elem が 1 より大きい場合や、VT で BSTR を指定した場合は無視されます。	FALSE: 配列型としない TRUE: 配列型とする	FALSE																						

表 3-6 I\_&lt;??&gt;, M\_&lt;??&gt;, O\_&lt;??&gt;変数の取得/設定データ型

VT オプション	Elem オプション	Array オプション	データ型
I1	1	FALSE	VT_I1
		TRUE	VT_I1   VT_ARRAY
	複数	FALSE	VT_I1   VT_ARRAY
		TRUE	VT_I1   VT_ARRAY
I2	1	FALSE	VT_I2
		TRUE	VT_I2   VT_ARRAY
	複数	FALSE	VT_I2   VT_ARRAY
		TRUE	VT_I2   VT_ARRAY
I4	1	FALSE	VT_I4
		TRUE	VT_I4   VT_ARRAY
	複数	FALSE	VT_I4   VT_ARRAY
		TRUE	VT_I4   VT_ARRAY
I8	1	FALSE	VT_I8
		TRUE	VT_I8   VT_ARRAY
	複数	FALSE	VT_I8   VT_ARRAY
		TRUE	VT_I8   VT_ARRAY
UI1	1	FALSE	VT_UI1
		TRUE	VT_UI1   VT_ARRAY
	複数	FALSE	VT_UI1   VT_ARRAY
		TRUE	VT_UI1   VT_ARRAY
UI2	1	FALSE	VT_UI2
		TRUE	VT_UI2   VT_ARRAY
	複数	FALSE	VT_UI2   VT_ARRAY
		TRUE	VT_UI2   VT_ARRAY
UI4	1	FALSE	VT_UI4
		TRUE	VT_UI4   VT_ARRAY
	複数	FALSE	VT_UI4   VT_ARRAY
		TRUE	VT_UI4   VT_ARRAY
UI8	1	FALSE	VT_UI8
		TRUE	VT_UI8   VT_ARRAY
	複数	FALSE	VT_UI8   VT_ARRAY

VT オプション	Elem オプション	Array オプション	データ型
		TRUE	VT_UI8   VT_ARRAY
R4	1	FALSE	VT_R4
		TRUE	VT_R4   VT_ARRAY
	複数	FALSE	VT_R4   VT_ARRAY
		TRUE	VT_R4   VT_ARRAY
R8	1	FALSE	VT_R8
		TRUE	VT_R8   VT_ARRAY
	複数	FALSE	VT_R8   VT_ARRAY
		TRUE	VT_R8   VT_ARRAY
BSTR	1	FALSE	VT_BSTR
		TRUE	VT_BSTR
	複数	FALSE	VT_BSTR
		TRUE	VT_BSTR

**使用例**

```
CaoController.AddVariable("I_000", "VT=UI2, Address=0, Elem=100")
```

**AREASIZE\_<??>**

AREASIZE\_<??>は指定されたエリアの容量をバイト単位で取得する変数です。<??>に任意の文字列を指定して使用してください。

表 3-7 AREASIZE\_&lt;??&gt;のオプション

オプション	必須	説明	設定できる値範囲		デフォルト値
Area=<タグ名称>	○	どのエリアの容量を取得するかを指定します。	0	入力エリア	---
			1	マーカエリア	
			2	出力エリア	

**使用例**

```
CaoController.AddVariable("AREASIZE_INPUT", "Area=0")
```

## 4. コマンドリファレンス

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

### 4.1. インスタンスリスト取得モード

---

#### GetRuntimeVersion コマンド

---

ランタイムバージョンを取得します。

引数	機能	
[in] vntParam	--	
[out] pVal	VT_BSTR	ランタイムバージョン(*.*)が格納されます。

---

#### GetInstances コマンド

---

シミュレーションに存在するすべてインスタンス情報を取得します。

引数	機能	
[in] vntParam	--	
[out] pVal	VT_VARIANT   VT_ARRAY	すべてのインスタンス情報が格納されます。
	i   VT_ARRAY   VT_VARIANT	インスタンス 1 つ分の情報が格納されます。
	0   VT_I4	ID が格納されます。
	1   VT_BSTR	インスタンス名が格納されます。

## 4.2. インスタンス制御モード

### UpdateTagList コマンド

接続しているインスタンスに定義されているタグリストを内部に取り込みます。

接続時に自動でタグリストは取り込まれます。

接続後に、別のソフトでインスタンスのタグリストを変更した際に呼び出してください。

引数	機能
[in] vntParam	--
[out] pVal	--

### SetPower コマンド

仮想コントローラの電源状態を設定します。

引数	機能
[in] vntParam	VT_UI1 電源状態設定値 0: 電源-OFF 1: 電源-ON
[out] pVal	--

### SetRun コマンド

仮想コントローラの CPU 状態を設定します。

引数	機能
[in] vntParam	VT_UI1 CPU 状態設定値 0: CPU-STOP 1: CPU-RUN
[out] pVal	--

### GetOperatingState コマンド

仮想コントローラの動作状態を取得します。

引数	機能
[in] vntParam	--
[out] pVal	VT_UI1 動作状態が格納されます。 1: 電源-OFF 2: 電源-ON 途中 3: CPU-STOP 4: CPU-RUN 移行中 5: CPU-RUN 6: CPU-フリーズ状態 7: 電源 OFF 途中

**GetCpuType コマンド**

仮想コントローラの CPU タイプを取得します。

引数	機能	
[in] vntParam	--	
[out] pVal	VT_I4	CPU 種別に対応する値が格納されます。

**GetTagInfos コマンド**

WString 型, String 型以外のすべてのタグ情報を読み込みます。

引数	機能	
[in] vntParam	--	
[out] pVal	VT_VARIANT   VT_ARRAY	すべてのインスタンス情報を格納します。
	i   VT_ARRAY   VT_VARIANT	インスタンス 1 つ分の情報を格納します。
	0   VT_BSTR	タグ名称が格納されます。
	1   VT_I4	型情報を表す VERENUM の値が格納されます。 タグの種類ごとにどの VERENUM が対応するかは表 3-4 を参照してください。

**GetTagNames コマンド**

WString 型, String 型以外のすべてのタグ名称を読み込みます。

引数	機能	
[in] vntParam	--	
[out] pVal	VT_BSTR   VT_ARRAY	すべてのタグ名称を格納します。

**GetType コマンド**

指定したタグの型情報を読み込みます。

引数	機能	
[in] vntParam	VT_BSTR	タグ名称 (大文字小文字は区別されます。)
[out] pVal	VT_I4	型情報を表す VERENUM の値が格納されます。

**GetAreaSize コマンド**

指定されたエリアのサイズを取得します。

ここで取得した値から 1 減算した値が I/M/O でアクセスできるアドレスの最大値となります。

引数	機能	
[in] vntParam	VT_UI1	入力エリアを指定します。 0: 入力エリア 1: マーカエリア 2: 出力エリア
[out] pVal	VT_I4	指定したエリアの最大容量 (バイト単位) が格納されます。

## 5. エラーコード

本プロバイダには、以下の独自エラーコードが存在します。（表 5-1 独自エラーコード表参照）

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 5-1 独自エラーコード表

エラー番号	説明
0x80110001	@INSTANCES未指定時にID, NAMEを指定していない.
0x80110002	CPU状態, 電源状態を取得できない状態.
0x80110003	存在しないタグ/体操していないタグ名を指定.
0x80110004	データ数と指定データ数が不一致.
0x80110005	指定されたエリアの範囲外を指定された.

## 6. サンプルプログラム

以下に INSTANCE のタグから値を取得する簡単なサンプルを示します。

前提条件：

- ・使用するインスタンス ID は 0 とする。
- ・タグ名は DataBlock.INT とする (タグのデータ型は VT\_I2)。

### List 3-1

### Sample.frm

```
Dim eng As CaoEngine
Dim ctrl As CaoController
Dim var As CaoVariable
Private Sub Form_Load()
    Set eng = New CaoEngine
    ' コントローラーオブジェクト作成
    Set ctrl = eng.Workspaces(0).AddController(
        "PLCSIM_1500",
        "CaoProv.SIEMENS.PLCSIM",
        "",
        "ID=0")
    ' コントローラー変数オブジェクト作成
    Set var = ctrl.AddVariable("TAG_DataBlock.INT", "Name=DataBlock.INT")
End Sub

' 値の取得
Private Sub Command1_Click()
    ' コントロール変数値の取得
    Dim meases As Variant
    Set meases = var.Value
    For i = 0 To UBound(meases)
        ' エクステンション変数オブジェクト作成
        Text[i].text = meases[i].get_Value
    Next i
End Sub
```