

SIEMENS PLCSIM provider

Version 1.1.0

User's guide

September 5, 2024

Remarks:

This document is translated from Japanese into English by the machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2016-09-05	First edition.
	2020-11-16	Added license additions.
1.1.0	2024-09-05	Corrected the part where character string release processing was not properly performed.

【 hardware 】

Model	Version	Notes
S7-PLCSIM Advanced	V1.0	

【Attention】

Additional license for "SIEMENS PLCSIM Provider" is required to use this provider.

Contents

1. Introduction.....	4
1.1. Installing license.....	5
2. Outline of provider	6
2.1. Outline	6
2.2. Method property.....	6
2.2.1. CaoWorkspace::AddController method	6
2.2.2. CaoController::AddVariable method	8
2.2.3. CaoController::get_VariableNames method	8
2.2.4. CaoController::Execute method.....	8
3. Variable reference	9
3.1. Instance list acquisition mode	9
3.2. Instance control mode.....	10
4. Command reference.....	22
4.1. Instance list acquisition mode	22
4.2. Instance control mode.....	23
5. Error code.....	26
6. Sample program.....	27

1. Introduction

This book is PLC simulation made of SIEMENS: It is a user's manual of the provider to control S7-PLC Simulation Advanced (simulation software and description). The simulation software can start by one or less a host, and can define two or more virtual controllers about one simulation set of software (It is called that it instances).

1.1. Installing license

To use this Provider, you need to install ORiN2 SDK, and also need to input “SIEMENS PLCSIM Provider” license information. If you would like to install it for evaluation, please use the following license.

SIEK-R15W-USE5-ETJB (valid for 3 months)

How to add the license is as follows.

1. Run the CaoConfig tool from the [Start] menu, and select the [Cao Provider] tab.
2. Select the [SIEMENS PLCSIM CAO Provider] item on the provider list.
3. Click the [...] button of the license input box.
4. Click the [Add] button in the “ORiN2 License Manager” window.
5. Input a license key, and click the [OK] button.
6. Click the [Close] button to exit.

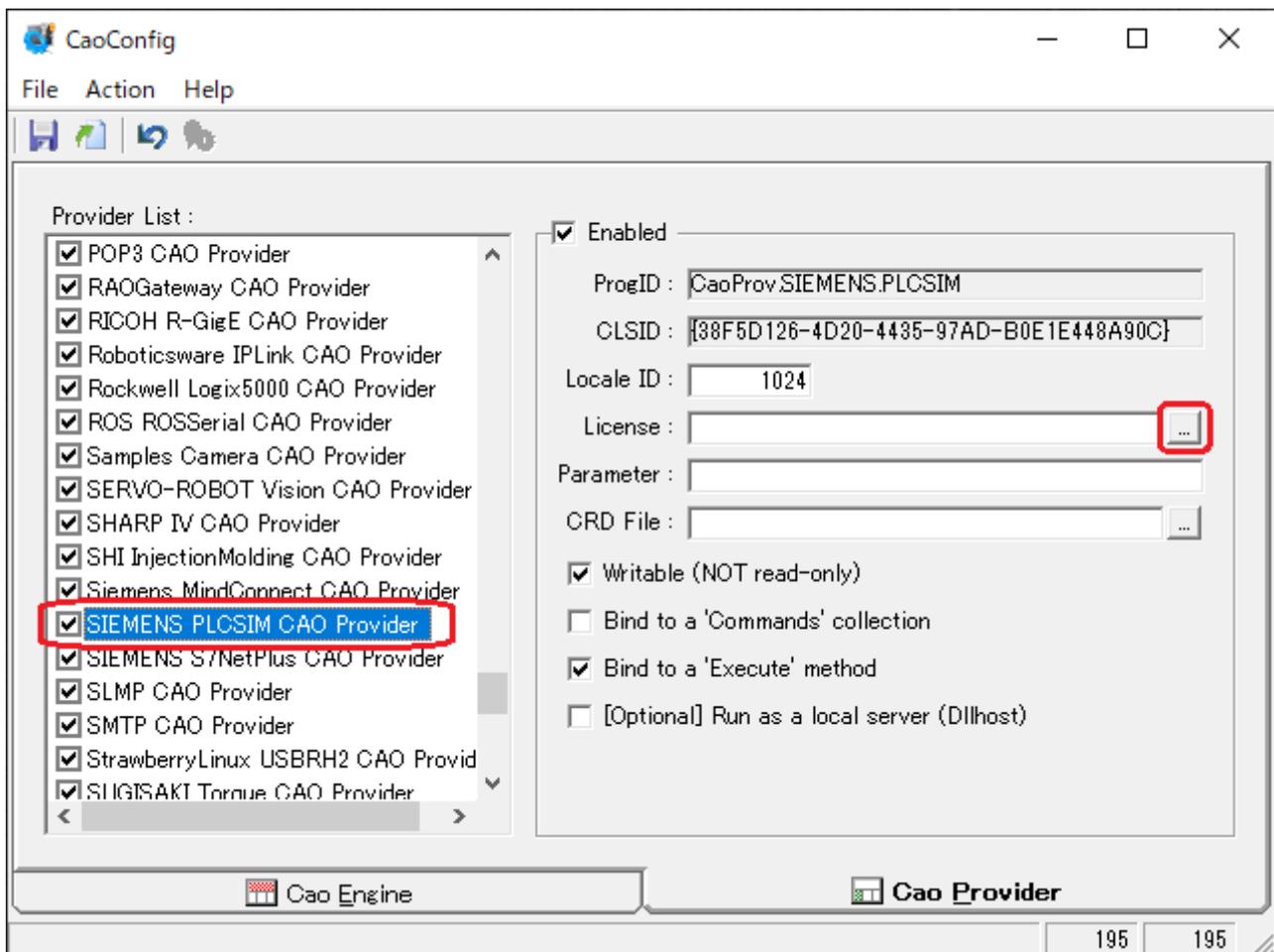


Figure 1-1 Installing ' SIEMENS PLCSIM Provider ' license

2. Outline of provider

2.1. Outline

The SIEMENS PLCSIM provider is DLL (Dynamic Link Library), please refer to Table 2-1 for the details.

Table 2-1 SIEMENS PLCSIM provider

File name	CaoProvSIEMENSPLCSIM.dll
ProgID	GaoProv.SIEMENS.PLCSIM
Registry registration	regsvr32 CaoProvSIEMENSPLCSIM.dll
Blotting out of registry registration	regsvr32 /u CaoProvSIEMENSPLCSIM.dll

There are roughly separately two operational modes in this provider.

1. Instance list acquisition mode

It is a mode to acquire the instance list that exists in the simulation software started by the specified host.

2. Instance control mode

It is a mode that controls the specified instance.

2.2. Method property

2.2.1. CaoWorkspace::AddController method

When the Controller object is generated, a necessary option is specified by the above-mentioned operational mode and each operational mode.

The specification of AddController is shown as follows.

Format

AddController

```
(
    "< controller name >",           // Controller name (arbitrariness)
    "GaoProv.SIEMENS.PLCSIM",       // Provider name (fixation)
    "< machine name >",             // Provider execution machine name (unused)
    "< option >"                   // Optional character string
)
```

The character string specified for an optional character string is shown as follows.

Table 2-2 Optional character string of CaoWorkspace::AddController

Option	Indispensability	Explanation	Range of value	Default value
@INSTANCES	--	It is specified whether it is an instance acquisition mode. When this option is specified, ID and optional Name are disregarded. When only the name is specified, this option is judged to be True.	True / False	True
Conn=<ETH option >	--	The host of the connection destination is specified. It accesses local host's simulation software when omitting it.	----	Dest IP: Indispensability Dest Port: 8192 Src IP: 255.255.255.255 Src Port: 0
ID=< instance ID >	--	ID of the instance for the control is specified. When @Instances is specified, it is disregarded.	0 -	0
Name=<Instance name of >	--	The name of the instance for the control is specified. When optional @Instances/Id is specified, it is disregarded.	----	

(usage example)

1. When you want to acquire the local host's instance list

@INSTANCES

2. When you want to control remote host's instance

Conn=Eth:10.4.9.24:8192, ID=0

3. When you want to control remote host's instance

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

2.2.2. CaoController::AddVariable method

The CaoVariable object is generated from CaoController. The variable of each operational mode that can be used is different.

The specification of AddVariable is shown as follows.

Format

AddVariable

```
(
    "< variable identifier >" // Variable identifier
    "" // Optional character string (unused)
)
```

2.2.3. CaoController::get_VariableNames method

Get variable name list is acquired.

Format

```
get_VariableNames
(
    <pVal:VARIANT*> // Out execution line result
)
```

2.2.4. CaoController::Execute method

Please refer to chapter 4 for the details of each command.

Format

```
Execute
(
    <bstrCommand:BSTR> // In command name
    [, <vntParam:VARIANT>] // In parameter
    [, <pVal:VARIANT>] // Out execution line result
)
```

3. Variable reference

3.1. Instance list acquisition mode

The variable list that can be used in the instance list acquisition mode is described. Only the get operation is possible by all variables.

Table 3-1 Controller class variable list that can be used in instance list acquisition mode

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	The manufacturer name is returned.	✓	-
@VERSION	VT_BSTR	The version of the provider is returned.	✓	-
@RUNTIMEVERSION	VT_BSTR	The run time version is returned.	✓	-
@Instances	VT_VARIANT VT_ARRAY	The list of all instances that exist in the simulation software is acquired.		
	i VT_VARIANT VT_ARRAY		✓	-
	0 VT_I4	ID is maintained.		
	1 VT_BSTR	The instance name is maintained.		

3.2. Instance control mode

Show the variable list that can be used in instance control mode.

Table 3-2 Controller class variable list that can be used in instance mode

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKERNAME	VT_BSTR	The manufacturer name is returned.	✓	-
@VERSION	VT_BSTR	The version of the provider is returned.	✓	-
@OPERATINGSTATE	VT_UI1	The virtual controller's operation is acquired. 1: Power supply OFF 2: Power supply ON on the way 3: CPU-STOP 4: CPU is shifting from STOP to RUN. 5: CPU-RUN 6: CPU-state of frieze 7: Power supply OFF on the way	✓	-
@POWER	VT_UI1	Virtual controller's state of CPU acquisition/is set. 0: Power supply- turning off 1: Power supply- turning on	✓	✓
@RUN	VT_UI1	Virtual controller's state of CPU acquisition/is set. 0: CPU-STOP 1: CPU-RUN	✓	✓
@ID	VT_I4	Instance ID is acquired.	✓	-
@NAME	VT_BSTR	The instance name is acquired.	✓	-
@CPUTYPE	VT_I4	Virtual controller's CPU kind is acquired.	✓	-
@TAGINFOS	VT_VARIANT VT_ARRAY	Tag information is read. However, the tag of the WString type and the String type is off the subject.	✓	-
	i VT_VARIANT VT_ARRAY			
	0 VT_I4	ID is maintained.		
	1 VT_BSTR	The instance name is maintained.		
@TAGNAMES	VT_BSTR VT_ARRAY	The tag name is read. However, the tag of the WString type and the String type is off the subject.	✓	-
TAG_<??>	It follows the type of tag.	The value of tag acquisition/is set.	✓	✓
I_<??>	It optional specification follows it.	The value that the specified address in the input area maintains acquisition/is set.	✓	✓
M_<??>	It optional specification follows it.	The value that the specified address in the marker area maintains acquisition/is set.	✓	✓
O_<??>	It optional specification follows it.	The value that the specified address in the output area maintains acquisition/is set.	✓	✓

AREASIZE_<??>	VT_UI4	Capacity (Byte) of the specified area is acquired. It becomes the maximum value of optional Address that can be specified in value I/M/O one subtraction from the value acquired in this variable.	✓	-
-------------------------------------	--------	--	---	---

TAG_<??>

TAG_<?? > is a variable to control the tag that exists in the instance. Please specify an arbitrary character string after TAG _. The tag of the WString type and the String type is excluded by limiting API used from this provider.

About other types, please refer to the Table3-2. When you arrange these types

Corresponding type| VT_ARRAY

It becomes it.

Table3-1 TAG_<?? > option

Option	Indispensability	Explanation	Range of value that can be set	Default value
Tag name of Name=< >	-	The tag name of the object that acquisition/sets the value is specified.	Character string	---

Please tie by "." and specify the hierarchy for the tag name. For instance, to control the Position element of the Robot structure that exists in DataBlock, it becomes "Name=DataBlock.Robot.Position".

Please use the @TAGNAMES variable to acquire the list of the tag name that can be specified and use the @TAGINFOS variable respectively when you want to acquire not only the tag name but also type information.

Table3-2 Correspondence table of type of tag and set acquisition of provider/type

Type in PLC simulation	Type in provider
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

Type in PLC simulation	Type in provider
DInt	VT_I4
UDInt	VT_UI4
Lint	VT_I8
ULInt	VT_UI8
Word	VT_UI2
Dword	VT_UI4
LWord	VT_UI8
Real	VT_R4
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	Uncorrespondence
WString	Uncorrespondence
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

Type in PLC simulation	Type in provider
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
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

Type in PLC simulation	Type in provider
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
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

Type in PLC simulation	Type in provider	
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
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

Type in PLC simulation	Type in provider
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
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

Type in PLC simulation	Type in provider
Struct	VT_ARRAY VT_VARIANT

Usage example

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

I_<??> / M_<??> / O_<??>

I_<??>/ M_<??>/ O_<??> is a variable that controls the value of each area of the instance. Please specify an arbitrary character string for > respectively. The corresponding area becomes the following.

- I_<??>: input area
- M_<??>: marker area
- O_<??>: output area

Table3-30option of O_<??> variable M_<??> .. I_<??> .. >

Option	Indispensability	Explanation	Range of value that can be set		Default value
VT=< type >	--	The type used is specified. Capital letters and small letters are not distinguished.	I1	One byte integer with sign	U11
			I2	Two byte integer with sign	
			I4	Four byte integer with sign	
			I8	Eight byte integer with sign	
			U1	One byte of sign none integer	
			U2	Two byte of sign none integer	
			U4	Four byte of sign none integer	
			U8	Eight byte of sign none integer	
			R4	Floating point	
			R8	Double-precision floating point	
			BSTR	Character string	
Address=< address >	--	The address where reading/writing is begun is specified by each byte.	$0 \leq Address \leq (AreaSize - 1)$		0

Option	Indispensability	Explanation	Range of value that can be set	Default value
Number of elements of Elem=< >	--	The number of elements is specified. However, when BSTR is specified for optional VT, it becomes the number of maximum characters.	$1 \leq Elem$ $\leq \frac{(AreaSize - Address)}{Sizeof(VT)}$	1
Array flag of Array=< >	--	Whether the value is acquired as an array type when Elem is one is specified. When Elem is larger than one, and BSTR is specified with VT, it is disregarded.	FALSE: It is not assumed the array type. TRUE: It is assumed the array type.	FALSE

Table3-4Acquisition/set data type of 0_<?? > variable M_<?? .. I_<?? >.. >

VT is optional.	Elem is optional.	Array is optional.	Data type
I1	1	FALSE	VT_I1
		TRUE	VT_I1 VT_ARRAY
	Plurals	FALSE	VT_I1 VT_ARRAY
		TRUE	VT_I1 VT_ARRAY
I2	1	FALSE	VT_I2
		TRUE	VT_I2 VT_ARRAY
	Plurals	FALSE	VT_I2 VT_ARRAY
		TRUE	VT_I2 VT_ARRAY
I4	1	FALSE	VT_I4
		TRUE	VT_I4 VT_ARRAY
	Plurals	FALSE	VT_I4 VT_ARRAY
		TRUE	VT_I4 VT_ARRAY
I8	1	FALSE	VT_I8
		TRUE	VT_I8 VT_ARRAY
	Plurals	FALSE	VT_I8 VT_ARRAY
		TRUE	VT_I8 VT_ARRAY
UI1	1	FALSE	VT_UI1
		TRUE	VT_UI1 VT_ARRAY
	Plurals	FALSE	VT_UI1 VT_ARRAY
		TRUE	VT_UI1 VT_ARRAY
UI2	1	FALSE	VT_UI2
		TRUE	VT_UI2 VT_ARRAY
	Plurals	FALSE	VT_UI2 VT_ARRAY
		TRUE	VT_UI2 VT_ARRAY
UI4	1	FALSE	VT_UI4
		TRUE	VT_UI4 VT_ARRAY
	Plurals	FALSE	VT_UI4 VT_ARRAY
		TRUE	VT_UI4 VT_ARRAY
UI8	1	FALSE	VT_UI8
		TRUE	VT_UI8 VT_ARRAY
	Plurals	FALSE	VT_UI8 VT_ARRAY

VT is optional.	Elem is optional.	Array is optional.	Data type
		TRUE	VT_UI8 VT_ARRAY
R4	1	FALSE	VT_R4
		TRUE	VT_R4 VT_ARRAY
	Plurals	FALSE	VT_R4 VT_ARRAY
		TRUE	VT_R4 VT_ARRAY
R8	1	FALSE	VT_R8
		TRUE	VT_R8 VT_ARRAY
	Plurals	FALSE	VT_R8 VT_ARRAY
		TRUE	VT_R8 VT_ARRAY
BSTR	1	FALSE	VT_BSTR
		TRUE	VT_BSTR
	Plurals	FALSE	VT_BSTR
		TRUE	VT_BSTR

Usage example

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

AREASIZE_<??>

AREASIE_<?? > is a variable to acquire the capacity of the specified area by each byte. Please use an arbitrary character string for > specifying it.

Table3-50option of AREASIZE_<?? >

Option	Indispensability	Explanation	Range of value that can be set		Default value
Tag name of Area=< >	-	The capacity of which area is acquired is specified.	0	Input area	---
			1	Marker area	
			2	Output area	

Usage example

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

4. Command reference

This chapter explains each command of the `CaoController::Execute` method.

4.1. Instance list acquisition mode

GetRuntimeVersion command

The run time version is acquired.

Argument	Function	
[in] vntParam	--	
[out] pVal	VT_BSTR	Run time version (*.*) is stored.

GetInstances command

All `..exist..` instance information is acquired the simulation.

Argument	Function	
[in] vntParam	--	
[out] pVal	VT_VARIANT VT_ARRAY	All instance information is stored.
	i VT_ARRAY VT_VARIANT	Information is stored for one instance minute.
	0 VT_I4	ID is stored.
	1 VT_BSTR	The instance name is stored.

4.2. Instance control mode

UpdateTagList command

The tag list defined in the connected instance is taken internally.

When connecting it, the tag list is taken by the automatic operation.

Please call it when you change the tag list of the instance with another software after it connects it.

Argument	Function
[in] vntParam	--
[out] pVal	--

SetPower command

Virtual controller's state of the power supply is set.

Argument	Function
[in] vntParam	VT_UI1 Power supply state setting value 0: Power supply- turning off 1: Power supply- turning on
[out] pVal	--

SetRun command

Virtual controller's state of CPU is set.

Argument	Function
[in] vntParam	VT_UI1 CPU state setting value 0: CPU-STOP 1: CPU-RUN
[out] pVal	--

GetOperatingState command

The virtual controller's operation is acquired.

Argument	Function
[in] vntParam	--
[out] pVal	VT_UI1 Operation is stored. 1: Power supply- turning off 2: It power supply-is turning it on. 3: CPU-STOP 4: CPU-RUN is shifting. 5: CPU-RUN 6: CPU-state of frieze 7: Power supply OFF on the way

GetCpuType command

Virtual controller's CPU type is acquired.

Argument	Function	
[in] vntParam	--	
[out] pVal	VT_I4	The value corresponding to CPU type is stored.

GetTagInfos command

All tag information except the WString type and the String type are read.

Argument	Function	
[in] vntParam	--	
[out] pVal	VT_VARIANT VT_ARRAY	All instance information is stored.
	i VT_ARRAY VT_VARIANT	Information is stored for one instance minute.
	0 VT_BSTR	The tag name is stored.
	1 VT_I4	The value of VERENUM that shows type information is stored. Which VERENUM of each kind of tag corresponds?Table3-2Please refer to [wo].

GetTagNames command

All the tag names except the WString type and the String type are read.

Argument	Function	
[in] vntParam	--	
[out] pVal	VT_BSTR VT_ARRAY	All the tag names are stored.

GetType command

Type information on the specified tag is read.

Argument	Function	
[in] vntParam	VT_BSTR	Tag name (Capital letters and small letters are distinguished).
[out] pVal	VT_I4	The value of VERENUM that shows type information is stored.

GetAreaSize command

The size of the specified area is acquired.

It becomes the maximum value of the address that can be accessed in value I/M/O one subtraction from the value acquired here.

Argument	Function	
[in] vntParam	VT_UI1	The input area is specified. 0: Input area 1: Marker area 2: Output area

[out] pVal	VT_I4	The maximum capacity of the specified area (unit of the byte) is stored.
------------	-------	--

5. Error code

In this provider, the following and original the error code exists. (Table5-1Reference)

About the ORiN2 commonness error, Please refer to the chapter of the error code of "ORiN2 programming guide".

Table5-1 Original error code table

Error number	Explanation
0x80110001	When @INSTANCES is unspecified, neither ID nor NAME are specified.
0x80110002	The state of CPU and the state of the power supply cannot be acquired.
0x80110003	The tag name that doesn't exist in which it tag/doesn't do physical exercise is specified.
0x80110004	The number of data and the number of specified data are the disagreements.
0x80110005	It was specified beyond the limits of the area specified.

6. Sample program

An easy sample to acquire the value from the tag of INSTANCE is shown as follows.

Precondition:

- Instance ID used is assumed to be 0.
- The tag name is assumed to be DataBlock.INT (The data type of tag is 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
    ' make controller object
    Set ctrl = eng.Workspaces(0).AddController(
        "PLCSIM_1500",
        "CaoProv.SIEMENS.PLCSIM",
        "",
        "ID=0")
    ' make Controller variable object
    Set var = ctrl.AddVariable("TAG_DataBlock.INT", "Name=DataBlock.INT")
End Sub

' Get value
Private Sub Command1_Click()
    ' Get control variable value
    Dim meases As Variant
    Set meases = var.Value
    For i = 0 To UBound(meases)
        Extension variable object ..'.. making
        Text[i].text = meases[i].get_Value
    Next i
End Sub

```