IAI Corporation
IAI SEL provider

Version 1.0.2

User's guide

December 21, 2021

Remarks:
This document uses the machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2017-09-22	First edition.
1.0.1	2018-10-31	The problem that the memory leak is generated when the connection fails is corrected.
	2020-10-6	Addition of license addition items. Typographical correction.
1.0.2	2021-12-21	Fixed false positives for errors.

【 hardware 】

Model	Version	Notes
SSEL	-	

【Attention】

Additional license for " IAI Program Type Controller Provider " is required to use this provider.

1. Introduction

This book is an user's guide of the SEL provider that acquires data from the SEL device.
It becomes easy to read and write data to the SEL device made by the IAI company if this provider is used.
This book explains the function of this SEL provider and the mounting method.

1.1. Installing license

To use OpenCV Provider, you need to install ORiN2 SDK, and also need to input “IAI Program Type Controller Provider” license information. If you would like to install it for evaluation, please use the following license.

IIEA-A262-RYCA-CR9G (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 [IAI SEL 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.

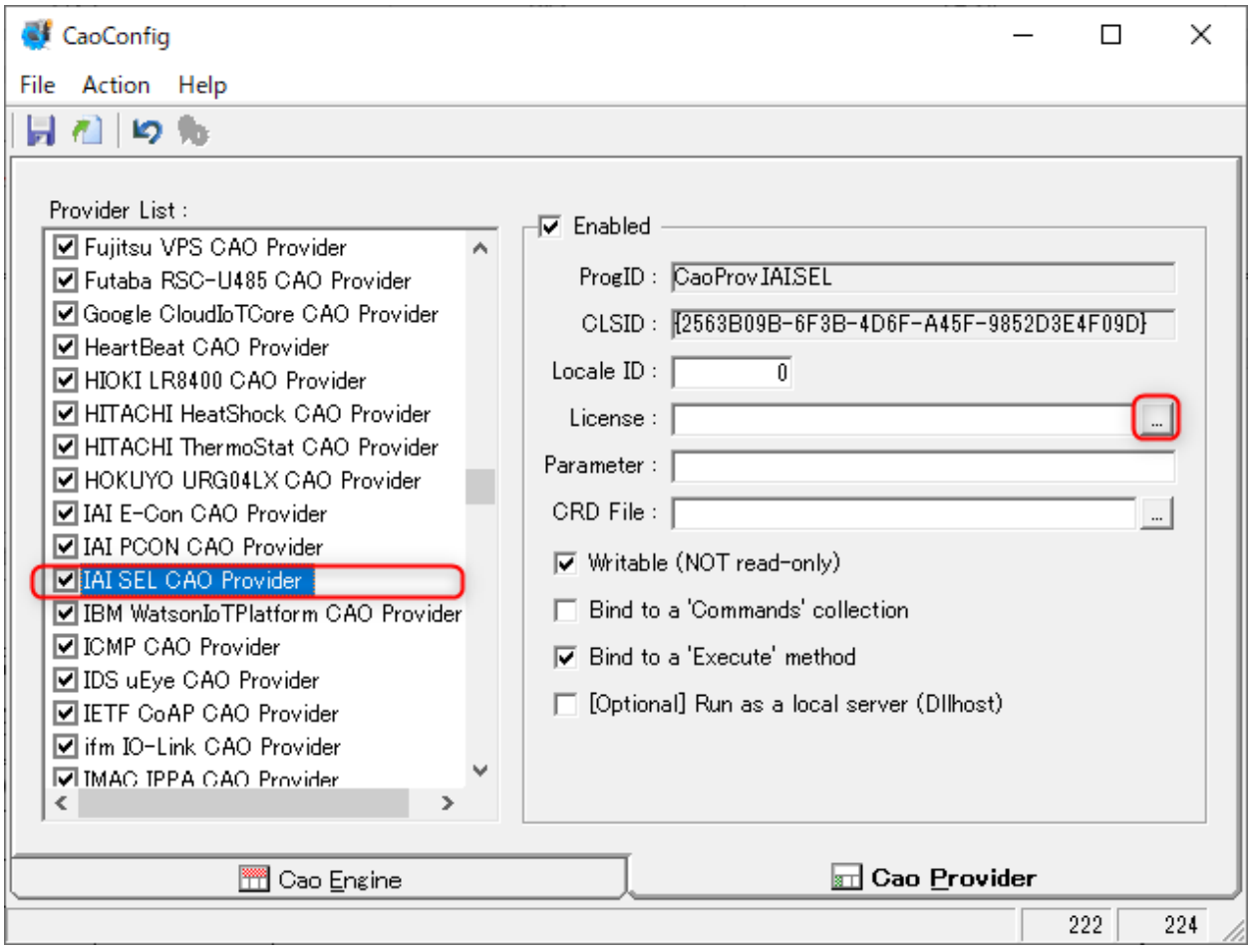

Figure 1-1 Installing ' IAI Program Type Controller Provider ' license

2. Outline of provider

2.1. Outline

The SEL provider reads and writes data according to the COM communication or the TCP/IP communication. The file format of the SEL provider is DLL(Dynamic Link Library), and please refer to Table2-1 for details.

Table2-1 SEL provider

File name	CaoProvIAISEL.dll
ProgID	CaoProv.IAI.SEL
Registration	regsvr32 CaoProvIAISEL.dll
Unregistration	regsvr32 /u CaoProvIAISEL.dll

Moreover, figure where correspondence each SEL device was shown the SEL provider is the under. Figure2-1 It becomes it.

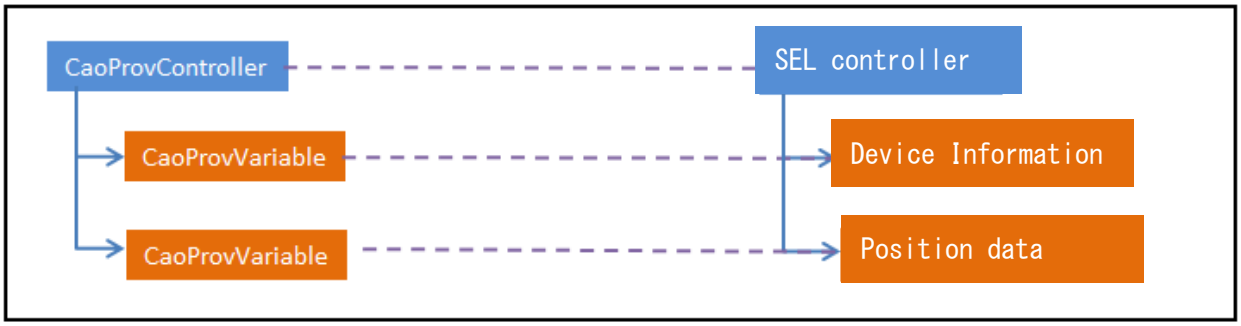

Figure2-1 Correspondence chart of CaoProvController and SEL device

2.2. About the axis pattern

The part in the value with the axis pattern and the description : the axis number of the object: It is two or more parameters that can be specified by the delimitation.

ex) 1:2:3:4:5:6:7:8

However, it becomes an error when setting it more than once.

ex) 1:2:3:4

2.3. Method property

2.3.1. CaoWorkspace::AddController method

When the Controller object is generated, a necessary option for the SEL device to connect it is specified.

The specification of AddController is shown as follows.

Format

```
AddController
(
        "< controller name >"   // Controller name (arbitrariness)
        "CaoProv.IAI.SEL",      // 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.

Table2-2Optional character string of CaoWorkspace::AddController

Option	Indispensability	Explanation	Range of value	Default value
CONN=COM:< number of COM port>	✓ *1	The COM port number of the connection destination is matched to the environment and it specifies it.	-----	-----
CONN=ETH:<connection destination IP>[:<number of Port>]	✓ *1	Internet Protocol address and the port of the connection destination are	-----	Port: 64611

		matched to the environment and it specifies it.		
TIMEOUT=<Response standby time>	-	The response standby time is specified (ms).	1 - 65535	500
STATION=< exchange number >	-	The exchange number of the device of the connection destination is matched to the environment and it specifies it.	0 - 255	99

＊1 Either of COM or ETH is indispensable.

ex)

```
Dim controller As Object
controller = Cao.AddController("SEL", "CaoProv.IAI.SEL", "", "CONN=COM:1,TIMEOUT=500,STATION=1")
```

2.3.2. CaoController:: Execute method

There is something that data can be read and written with CaoController in using Execute. The specification of Execute is shown as follows.

Format

```
Execute
(
        "< method name >"       // Method name
        "< argument >"          // Argument
)
```

Table2-3CaoController::Execute method list

Command name	Explanation	Link
TEST_CALL	The test of communications is executed.	P. 14
VERSION_CODE	The acquisition of the version code is executed.	P. 14
POSITION_DATA_NUM	The acquisition of the number of effective position data is executed.	P. 15
POSITION_DATA	The acquisition of the effective position data is executed.	P. 15
INPUT_PORT	The acquisition of the input port is executed.	P. 16
OUTPUT_PORT	The acquisition of the output port is executed.	P. 16
FLAG	The acquisition of the flag is executed.	P. 17
INTEGER_VARIABLE	The acquisition of the integer variable is executed.	P. 17
REAL_VARIABLE	The acquisition of the real number variable is executed.	P. 18
STRING_VARIABLE	The acquisition of the string variable is executed.	P. 18
AXIS_STATUS	The acquisition of the axis status is executed.	P. 19
PROGRAM_STATUS	The acquisition of the program status is executed.	P. 19
SYSTEM_STATUS	The acquisition of the system status is executed.	P. 20
ERROR_INFO	The acquisition of error details information is executed.	P. 21
POSITION_DATA_NUM2	The acquisition of number 2 of effective position data is executed.	P. 22

2.3.2.1. CaoController::Execute("TEST_CALL") command

The test of communications is executed.

Argument:

VT_BSTR	The transmitted character string is specified.

Return value:

VT_BSTR	The received character string is returned. (What specified by the transmitted character string and the same one are returned.)

Usage example:

```
Dim sVal As String
sVal = controller.Execute("TEST_CALL" , "01234567489")
```

2.3.2.2. CaoController::Execute("VERSION_CODE") command

The acquisition of the version code is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Inquiry head position No. is specified.
1	VT_UI2	The number of inquiry records is specified.
2	VT_UI2	The number of effective axes is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	The model code is returned.
1	VT_UI2	The unit code is returned.
2	VT_UI2	Version No. is returned.
3	VT_UI2	Time is returned.
4	VT_UI2	Time is returned.
5	VT_UI2	Time is returned.
6	VT_UI2	Time is returned.
7	VT_UI2	Time is returned.
8	VT_UI2	Time is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("VERSION_CODE" , Array(1, 2, 3))
```

2.3.2.3. CaoController::Execute("POSITION_DATA_NUM ") command

The acquisition of the number of effective position data is executed.

The argument: It is not.

Return value:

VT_UI2	The number of effective position data is returned.

Usage example:

```
Dim lVal As Long
lVal = controller.Execute("POSITION_DATA_NUM ")
```

2.3.2.4. CaoController::Execute("POSITION_DATA ") command

The acquisition of the effective position data is executed.

Argument:

VT_ARRAY | VT_UI2			
	0	VT_UI2	Inquiry head position No. is specified.
	1	VT_UI2	The number of inquiry records is specified.
	2	VT_UI2	The number of effective axes is specified.

Return value:

VT_ARRAY | VT_VARIANT					
	0	VT_ARRAY | VT_VARIANT			
		0	VT_UI2		Position No. is returned.
		1	VT_BSTR		The axis pattern is returned.
		2	VT_UI2		The acceleration is returned.
		3	VT_UI2		The deceleration is returned.
		4	VT_UI2		The speed is returned.
		5	VT_ARRAY | VT_I4		
			n	VT_I4	The positional data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA", Array(1, 2, 3))
```

2.3.2.5. CaoController::Execute("INPUT_PORT ") command

The acquisition of the input port is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Inquiry beginning port No. is specified.
1	VT_UI2	The number of inquiry ports is specified.

Return value:

VT_ARRAY | VT_UI2		
n	VT_UI2	The input port data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("INPUT_PORT", Array(1, 2))
```

2.3.2.6. CaoController::Execute("OUTPUT_PORT") command

The acquisition of the output port is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Inquiry beginning port No. is specified.
1	VT_UI2	The number of inquiry ports is specified.

Return value:

Usage example:

VT_ARRAY | VT_UI2		
n	VT_UI2	The output port data is returned.

```
Dim vArray As Variant
vArray = controller.Execute("OUTPUT_PORT", Array(1, 2))
```

2.3.2.7. CaoController::Execute("FLAG") command

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Beginning flag No. is specified.
2	VT_UI2	The number of flags is specified.
3	VT_BOOL	The array existence is specified. (The flag data is returned with VT_UI2 in case of true and the number of of flag =1.)

Return value:

VT_ARRAY | VT_UI2		
n	VT_UI2	The flag data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("FLAG", Array(1, 2, 3, true))
```

2.3.2.8. CaoController::Execute("INTEGER_VARIABLE") command

The acquisition of the integer variable is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Beginning variable No. is specified.
2	VT_UI2	The number of variable data is specified.
3	VT_BOOL	The array existence is specified. (The integer variable data is returned with VT_UI2 in case of true and the variable data the number of =1.)

Return value:

VT_ARRAY | VT_I4		
n	VT_I4	The integer variable data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("INTEGER_VARIABLE", Array(1, 2, 3, true))
```

2.3.2.9. CaoController::Execute("REAL_VARIABLE ") command

The acquisition of the real number variable is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Beginning variable No. is specified.
2	VT_UI2	The number of variable data is specified.
3	VT_BOOL	The array existence is specified. (The integer variable data is returned with VT_UI2 in case of true and the variable data the number of =1.)
4	VT_UI2	Endian is specified.

Return value:

VT_ARRAY | VT_R8		
n	VT_R8	The real number variable data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("REAL_VARIABLE")
```

2.3.2.10. CaoController::Execute("STRING_VARIABLE ") command

The acquisition of the string variable is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Beginning variable No. is specified.
2	VT_UI2	The number of variable data is specified.

Return value:

VT_BSTR	The string variable data is returned.

Usage example:

```
Dim sVal As String
sVal = controller.Execute("STRING_VARIABLE ", Array(1, 2 ,3))
```

2.3.2.11. CaoController::Execute("AXIS_STATUS ") command

The acquisition of the axis status is executed.

Argument:

VT_BSTR	The inquiry axis pattern is specified.

Return value:

VT_ARRAY | VT_VARIANT				
	0	VT_ARRAY | VT_I4		
		0	VT_I4	The axis status is returned.
		1	VT_I4	The axis sensor input status is returned.
		2	VT_I4	The error code related to the axis is returned.
		3	VT_I4	The encoder status is returned. (Reset it.)
		4	VT_I4	Present location is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("AXIS_STATUS", "1:2:3")
```

2.3.2.12. CaoController::Execute("PROGRAM_STATUS ") command

The acquisition of the program status is executed.

Argument:

VT_UI2	Program No. is specified.

Return value:

VT_ARRAY | VT_UI2			
	0	VT_UI2	Status is returned.
	1	VT_UI2	Program step No. is returned while executing it.
	2	VT_UI2	The program dependence error code is returned.
	3	VT_UI2	Error generation step No. is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("PROGRAM_STATUS", 1)
```

2.3.2.13. CaoController::Execute("SYSTEM_STATUS ") command

The acquisition of the system status is executed.

The argument: It is not.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	System mode
1	VT_UI2	Heavy level system error No.
2	VT_UI2	Latest system error No.
3	VT_UI2	System status ..byte.. one
4	VT_UI2	System status ..byte.. two
5	VT_UI2	System status ..byte.. three
6	VT_UI2	System status ..byte.. four

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("SYSTEM_STATUS")
```

2.3.2.14. CaoController::Execute("ERROR_INFO ") command

The acquisition of error details information is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Type 1 is specified.
1	VT_UI2	Type 2 is specified.
2	VT_UI2	Record No. is specified.

Return value:

VT_ARRAY | VT_VARIANT		
0	VT_UI4	Error No. is returned.
1	VT_UI4	Detailed information 1 is returned.
2	VT_UI4	Detailed information 2 is returned.
3	VT_UI4	Detailed information 3 is returned.
4	VT_I4	Detailed information 4 is returned.
5	VT_UI4	Detailed information 5 is returned.
6	VT_UI4	Detailed information 6 is returned.
7	VT_UI4	Detailed information 7 is returned.
8	VT_UI4	Detailed information 8 is returned.
9	VT_BSTR	The message character string is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("ERROR_INFO", Array(1, 2, 3))
```

2.3.2.15. CaoController::Execute("POSITION_DATA_NUM 2") command

The acquisition of number 2 of effective position data is executed.

The argument: It is not.

Return value:

VT_UI2	The number of effective position data is returned.

Usage example:

```
Dim lVal As Long
lVal = controller.Execute("POSITION_DATA_NUM2")
```

2.3.2.16. CaoController::Execute("POSITION_DATA 2") command

The acquisition of effective position data 2 is executed.

Argument:

VT_ARRAY | VT_UI2			
	0	VT_UI2	Inquiry head position No. is specified.
	1	VT_UI2	The number of inquiry records is specified.
	2	VT_UI2	The number of effective axes is specified.

Return value:

VT_ARRAY | VT_VARIANT					
	0	VT_ARRAY | VT_VARIANT			
		0	VT_UI2		Position No. is returned.
		1	VT_BSTR		The axis pattern is returned.
		2	VT_UI2		The acceleration is returned.
		3	VT_UI2		The deceleration is returned.
		4	VT_UI2		The speed is returned.
		5	VT_ARRAY | VT_I4		
			n	VT_I4	The positional data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA2", Array(1, 2, 3))
```

2.3.2.17. CaoController::Execute("POSITION_DATA3") command

The acquisition of effective position data 3 is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	The record format extension specification type is specified.
1	VT_UI2	Inquiry head position No. is specified.
2	VT_UI2	The number of inquiry records is specified.
3	VT_UI2	The number of effective axes is specified.

Return value:

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_VARIANT		
	0	VT_UI2	Position No. is returned.
	1	VT_UI2	The axis pattern is returned.
	2	VT_UI2	The acceleration is returned.
	3	VT_UI2	The deceleration is returned.
	4	VT_UI2	The speed is returned.
	5	VT_ARRAY | VT_I4	
		n VT_I4	The positional data is returned.
	6	VT_UI2	The enhancing data is returned.
	7	VT_UI4	The output function code is returned.
	8	VT_UI4	Output port flag No. is returned.
	9	VT_UI4	Function parameter 1 is returned.
	10	VT_UI4	Function parameter 2 is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("POSITION_DATA3", Array(1, 2, 3, 4))
```

2.3.2.18. CaoController::Execute("SERVO ") command

On/off of the servo is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	The operation type is specified.

The return value: It is not.

Usage example:

```
controller.Execute("SERVO", Array("1:2:3", 1))
```

2.3.2.19. CaoController::Execute("ORIGIN ") command

The starting point return is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	When the starting point returns, the end search speed is specified.
2	VT_UI2	When the starting point returns, the creep velocity is specified.

The return value: It is not.

Usage example:

```
controller.Execute("ORIGIN", Array("1:2:3", 2, 3))
```

2.3.2.20. CaoController::Execute("MOVE_ABSOLUITE") command

The coordinates specification movement is absolutely executed.

Argument:

VT_ARRAY | VT_VARIANT				
0	VT_BSTR			The axis pattern is specified.
1	VT_UI2			The acceleration is specified.
2	VT_UI2			The deceleration is specified.
3	VT_UI2			The speed is specified.
4	VT_ARRAY | VT_I4			
		n	VT_I4	Coordinate data is absolutely specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_ABSOLUITE", Array("1:2:3", 2, 3, 4, Array(10, 20)))
```

2.3.2.21. CaoController::Execute("MOVE_RELATIVE ") command

The relative coordinates specification movement is executed.

Argument:

VT_ARRAY | VT_VARIANT				
0	VT_BSTR			The axis pattern is specified.
1	VT_UI2			The acceleration is specified.
2	VT_UI2			The deceleration is specified.
3	VT_UI2			The speed is specified.
4	VT_ARRAY | VT_I4			
		n	VT_I4	Relative coordinate data is specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_RELATIVE", Array("1:2:3", 2, 3, 4, Array(10, 20)))
```

2.3.2.22. CaoController::Execute("MOVE_JOG_INCHUNG ") command

The [jogu] inching movement is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	The acceleration is specified.
2	VT_UI2	The deceleration is specified.
3	VT_UI2	The speed is specified.
4	VT_UI4	The inching distance is specified.
5	VT_UI2	The operation type is specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_JOG_INCHUNG", Array("1:2:3", 1, 2, 3, 4, 1))
```

2.3.2.23. CaoController::Execute("MOVE_POSITION_NO ") command

The position No. specification movement is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	The acceleration is specified.
2	VT_UI2	The deceleration is specified.
3	VT_UI2	The speed is specified.
4	VT_UI2	Position No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_POSITION_NO", Array("1:2:3", 1, 2, 3, 4))
```

2.3.2.24. CaoController::Execute("STOP_CANCEL ") command

Operation stop & cancellation is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The stop axis pattern is specified.
1	VT_UI2	The addition command byte is specified.

The return value: It is not.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("STOP_CANCEL", Array("1:2:3", 2))
```

2.3.2.25. CaoController::Execute("WRITE_POSITION_DATA_RANGE") command

A position data range specification and continuous writing is executed.

Argument:

VT_ARRAY | VT_VARIANT			
0	VT_UI2		Change beginning position data No. is specified.
1	VT_UI2		The number of change position data is specified.
2	VT_ARRAY | VT_VARIANT		
	0	VT_BSTR	The axis pattern is specified.
	1	VT_UI2	The acceleration is specified.
	2	VT_UI2	The deceleration is specified.
	3	VT_UI2	The speed is specified.
	4	VT_ARRAY | VT_I4	
		n VT_I4	The positional data is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	Change beginning position data No. is returned.
1	VT_UI2	The number of change completion position data is returned.

Usage example:

```
Dim vArray As Variant
```

```
vArray = controller.Execute("WRITE_POSITION_DATA_RANGE", Array(1, 2, Array("1:2:3", 4, 5, 6,
Array(10, 20)))
```

2.3.2.26. CaoController::Execute("WRITE_CHANGE_POSITION_DATA ") command

A change position data continuous writing is executed.

Argument:

VT_ARRAY | VT_VARIANT						
0	VT_UI2					The number of change position data is specified.
1	VT_ARRAY | VT_VARIANT					
	0	VT_UI2				Change position data No. is specified.
	1	VT_BSTR				The axis pattern is specified.
	2	VT_UI2				The acceleration is specified.
	3	VT_UI2				The deceleration is specified.
	4	VT_UI2				The speed is specified.
	5	VT_ARRAY | VT_I4				
			n	VT_I4		The positional data is specified.

Return value:

VT_UI2	The number of change completion position data is returned.

Usage example:

```
Dim lVal As Long
lVal = controller.Execute("WRITE_CHANGE_POSITION_DATA", Array(1, Array(2, "1:2:3", 4, 5, 6,
Array(10, 20))))
```

2.3.2.27. CaoController::Execute("CLEAR_POSITION_DATA ") command

Position clear data is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Clear beginning position data No. is specified.
1	VT_UI2	The number of clear position data is specified.

The return value: It is not.

Usage example:

controller.Execute("CLEAR_POSITION_DATA", Array(1, 2))

2.3.2.28. CaoController::Execute("CHANGE_OUTPUT_PORT_STATUS ") command

The output port state change is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Output port No. is specified.
1	VT_UI2	The change type is specified.

The return value: It is not.

Usage example:

controller.Execute("CHANGE_OUTPUT_PORT_STATUS ", Array(1, 1))

2.3.2.29. CaoController::Execute("CHANGE_FLUG_STATUS ") command

The flag state change is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Flag No. is specified.
2	VT_UI2	The change type is specified.

The return value: It is not.

Usage example:

controller.Execute("CHANGE_FLUG_STATUS", Array(1, 2, 1))

2.3.2.30. CaoController::Execute("CHANGE_INTEGER_VARIABLE") command

The integer variable change is executed.

Argument:

VT_ARRAY | VT_VARIANT				
0	VT_UI2			Program No. is specified.
1	VT_UI2			Change beginning variable No. is specified.
2	VT_UI2			The number of change variable data is specified.
3	VT_ARRAY | VT_I4			
		n	VTI4	The integer variable data is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	Program No. is returned.
1	VT_UI2	Change beginning variable No. is returned.
2	VT_UI2	The number of change completion data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("CHANGE_INTEGER_VARIABLE", Array(1, 2, 3, Array(10, 20, 30)))
```

2.3.2.31. CaoController::Execute("CHANGE_REAL_VARIABLE ") command

The real number variable change is executed.

Argument:

VT_ARRAY | VT_VARIANT			
0	VT_UI2		Program No. is specified.
1	VT_UI2		Change beginning variable No. is specified.
2	VT_UI2		The number of change variable data is specified.
3	VT_ARRAY | VT_R8		
	n	VT_R8	The real number variable data is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	Program No. is returned.
1	VT_UI2	Change beginning variable No. is returned.
2	VT_UI2	The number of change completion data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("CHANGE_REAL_VARIABLE", Array(1, 2, 3, Array(10, 20, 30)))
```

2.3.2.32. CaoController::Execute("CHANGE_STRING_VARIABLE") command

The string variable change is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_UI2	Program No. is specified.
1	VT_UI2	Change beginning variable No. is specified.
2	VT_UI2	The number of change variable data is specified.
3	VT_BSTR	The string variable data is specified.

The return value: It is not.

Usage example:

```
controller.Execute("CHANGE_STRING_VARIABLE", Array(1, 2, 3, "ABC"))
```

2.3.2.33. CaoController::Execute("RESET_ALARM ") command

Alarm reset is executed.

The argument: It is not.

The return value: It is not.

Usage example:

```
controller.Execute("RESET_ALARM")
```

2.3.2.34. CaoController::Execute("EXECUTION_PROGRAM ") command

The program execution is executed.

Argument:

VT_UI2	Program No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("EXECUTION_PROGRAM", 1)
```

2.3.2.35. CaoController::Execute("END_PROGRAM ") command

The program end is executed.

Argument:

VT_UI2	Program No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("END_PROGRAM", 1)
```

2.3.2.36. CaoController::Execute("PAUSE_PROGRAM ") command

The stop is executed at one o'clock of the program.

Argument:

VT_UI2	Program No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("PAUSE_PROGRAM", 1)
```

2.3.2.37. CaoController::Execute("EXECUTION_ONE_STEP_PROGRAM ") command

One program step execution is executed.

Argument:

VT_UI2	Program No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("EXECUTION_ONE_STEP_PROGRAM", 1)
```

2.3.2.38. CaoController::Execute("EXECUTION_RESUME_PROGRAM ") command

The program execution restart is executed.

Argument:

VT_UI2	Program No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("EXECUTION_RESUME_PROGRAM", 1)
```

2.3.2.39. CaoController::Execute("RESET_SOFTWARE ") command

Software reset is executed.

The argument: It is not.

The return value: It is not.

Usage example:

```
controller.Execute("RESET_SOFTWARE")
```

2.3.2.40. CaoController::Execute("REQUEST_RESTORATION_DRIVE ") command

Driving source restoration demand is executed.

The argument: It is not.

The return value: It is not.

Usage example:

```
controller.Execute("REQUEST_RESTORATION_DRIVE")
```

2.3.2.41. CaoController::Execute("REQUEST_PAUSE_RELEASE_OPERATION ") command

The stop release demand is executed at one o'clock of operation.

The argument: It is not.

The return value: It is not.

Usage example:

```
controller.Execute("REQUEST_PAUSE_RELEASE_OPERATION")
```

2.3.2.42. CaoController::Execute("CHANGE_SPEED") command

The speed change is executed.

Argument:

VT_ARRAY | VT_VARIANT			
	0	VT_BSTR	The axis pattern is specified.
	1	VT_UI2	The speed is specified.

The return value: It is not.

Usage example:

```
controller.Execute("CHANGE_SPEED", Array("1:2:3", 2))
```

2.3.2.43. CaoController::Execute("MOVE_POSITION_NO2") command

Position No. specification movement 2 is executed.

Argument:

VT_ARRAY | VT_VARIANT			
	0	VT_BSTR	The axis pattern is specified.
	1	VT_UI2	The acceleration is specified.
	2	VT_UI2	The deceleration is specified.
	3	VT_UI2	The speed is specified.
	4	VT_UI2	Position No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_POSITION_NO2", Array("1:2:3", 2, 3, 4, 5))
```

2.3.2.44. CaoController::Execute("WRITE_POSITION_DATA_RANGE2") command

Position data range specification and continuous writing 2 is executed.

Argument:

VT_ARRAY | VT_VARIANT						
0	VT_UI2					Change beginning position data No. is specified.
1	VT_UI2					The number of change position data is specified.
2	VT_ARRAY | VT_VARIANT					
	0	VT__BSTR				The axis pattern is specified.
	1	VT_UI2				The acceleration is specified.
	2	VT_UI2				The deceleration is specified.
	3	VT_UI2				The speed is specified.
	4	VT_ARRAY | VT_I4				
			n	VT_I4		The positional data is specified.

The return value: It is not.

Usage example:

controller.Execute("WRITE_POSITION_DATA_RANGE2", Array(1, 1, Array("1:2:3", 2, 3, 4, Array(10, 20))))

2.3.2.45. CaoController::Execute("WRITE_CHANGE_POSITION_DATA2") command

Change position data continuous writing 2 is executed.

Argument:

VT_ARRAY | VT_VARIANT				
0	VT_UI2			The number of change position data is specified.
1	VT_ARRAY | VT_VARIANT			
	0	VT_UI2		Change position data No. is specified.
	1	VT_BSTR		The axis pattern is specified.
	2	VT_UI2		The acceleration is specified.
	3	VT_UI2		The deceleration is specified.
	4	VT_UI2		The speed is specified.
	5	VT_ARRAY | VT_I4		
		n	VT_I4	The positional data is specified.

Return value:

VT_UI2	The number of change completion position data is returned.

2.3.2.46. CaoController::Execute("CLEAR_POSITION_DATA2") command

Position data clearness 2 is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Clear beginning position data No. is specified.
1	VT_UI2	The number of clear position data is specified.

The return value: It is not.

Usage example:

```
call controller.Execute("CLEAR_POSITION_DATA2", Array(1,2))
```

2.3.2.47. CaoController::Execute("CONTROLLER_FUNCTION2") command

Controller function specification 2 is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Controller function specification word 9 is specified.
1	VT_UI2	Controller function specification word 10 is specified.
2	VT_UI2	Controller function specification word 11 is specified.
3	VT_UI2	Controller function specification word 12 is specified.
4	VT_UI2	Controller function specification word 13 is specified.
5	VT_UI2	Controller function specification word 14 is specified.
6	VT_UI2	Controller function specification word 15 is specified.
7	VT_UI2	Controller function specification word 16 is specified.

The return value: It is not.

Usage example:

```
controller.Execute("CONTROLLER_FUNCTION2", Array(1, 2, 3, 4, 5, 6, 7, 8))
```

2.3.2.48. CaoController::Execute("WRITE_POSITION_DATA_RANGE3") command

Position data range specification and continuous writing 3 is executed.

Argument:

VT_ARRAY | VT_VARIANT			
0	VT_UI2		The record format extension specification type is specified.
1	VT_UI2		Change beginning position data No. is specified.
2	VT_UI2		The number of change position data is specified.
3	VT_ARRAY | VT_VARIANT		
	0	VT_UI2	The axis pattern is specified.
	1	VT_UI2	The acceleration is specified.
	2	VT_UI2	The deceleration is specified.
	3	VT_UI2	The speed is specified.
	4	VT_ARRAY | VT_I4	
		n VT_I4	The positional data is specified.
	5	VT_I4	The enhancing data is specified.
	6	VT_I4	The output function code is specified.
	7	VT_I4	Output port flag No. is specified.
	8	VT_I4	Function parameter 1 is specified.
	9	VT_I4	Function parameter 2 is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	The record format extension specification type is returned.
1	VT_UI2	Change beginning position data No. is returned.
2	VT_UI2	The number of change completion position data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_POSITION_DATA_RANGE3", Array(1 ,2, 1, Array("1:2:3", 2, 3, 4, Array(10, 20), 1, 2, 3, 4, 5)))
```

2.3.2.49. CaoController::Execute("WRITE_CHANGE_POSITION_DATA3") command

Change position data continuous writing 3 is executed.

Argument:

VT_ARRAY | VT_VARIANT				
0	VT_UI2			The record format extension specification type is specified.
1	VT_UI2			The number of change position data is specified.
2	VT_ARRAY | VT_VARIANT			
	0	VT_UI2		Change position data No. is specified.
	1	VT_UI2		The axis pattern is specified.
	2	VT_UI2		The acceleration is specified.
	3	VT_UI2		The deceleration is specified.
	4	VT_UI2		The speed is specified.
	5	VT_ARRAY | VT_I4		
		n	VT_I4	The positional data is specified.
	6	VT_I4		The enhancing data is specified.
	7	VT_I4		The output function code is specified.
	8	VT_I4		Output port flag No. is specified.
	9	VT_I4		Function parameter 1 is specified.
	10	VT_I4		Function parameter 2 is specified.

Return value:

VT_ARRAY | VT_UI2		
0	VT_UI2	The record format extension specification type is returned.
1	VT_UI2	The number of change completion position data is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("WRITE_CHANGE_POSITION_DATA3", Array(1, 1, Array(1, "1:2:3", 2, 3, 4, Array(10, 20), 1, 2, 3, 4, 5))
```

2.3.2.50. CaoController::Execute("COORDINATE_DATA_RANGE ") command

Acquisition for coordinate system definition data range continuous specification is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	The type is specified.
1	VT_UI2	Head coordinate system definition data No. of the inquiry distant future is specified.
2	VT_UI2	The number of inquiry records is specified.

Return value:

VT_ARRAY | VT_VARIANT				
0	VT_ARRAY | VT_I4			
		0	VT_I4	The amount of X coordinates offset is returned.
		1	VT_I4	The amount of Y coordinates offset is returned.
		2	VT_I4	The amount of Z coordinates offset is returned.
		3	VT_I4	The amount of R coordinates offset is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("COORDINATE_DATA_RANGE", Array(1, 2, 3))
```

2.3.2.51. CaoController::Execute("UNIT_AXIS_STATUS ") command

The acquisition of the unit axis status is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The inquiry axis pattern is specified.
1	VT_UI2	The type is specified.

Return value:

VT_ARRAY | VT_VARIANT			
0	VT_UI2		Work coordinate system selection No. is returned.
1	VT_UI2		Tool coordinate system selection No. is returned.
2	VT_UI2		Axis common status is returned.
3	VT_ARRAY | VT_I4		
	0	VT_I4	The axis status is returned.
	1	VT_I4	The axis sensor input status is returned.
	2	VT_I4	The error code related to the axis is returned.
	3	VT_I4	The encoder status (Reset it) is returned.
	4	VT_I4	Present location is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("UNIT_AXIS_STATUS", Array("1:2:3", 1))
```

2.3.2.52. CaoController::Execute("CHECK_ZONE_DATA_RANGE ") command

Acquisition for simple interference check zone definition data range continuous specification is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_UI2	Inquiry head and simplicity interference check zone definition data No. is specified.
1	VT_UI2	The number of inquiry records is specified.
2	VT_UI2	The number of effective axes is specified.

Return value:

VT_ARRAY | VT_VARIANT				
0	VT_ARRAY | VT_VARIANT			
	0	VT_UI2		The simple interference check zone definition coordinates effective axis pattern is returned.
	1	VT_ARRAY | VT_I4		
		n	VT_I4	Simple interference check zone definition coordinates 1 are returned.
	2	VT_ARRAY | VT_I4		
		n	VT_I4	Simple interference check zone definition coordinates 2 are returned.
	3	VT_UI2		Outputting physical output port No. when invading Or global flag No. is returned.
	4	VT_UI2		When invading, the error type specification is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("CHECK_ZONE_DATA_RANGE", Array(1, 2, 3))
```

2.3.2.53. CaoController::Execute("UNIT_AXIS_STATUS2") command

The acquisition of unit axis status 2 is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The inquiry axis pattern is specified.
1	VT_UI2	The type is specified.

Return value:

VT_ARRAY | VT_VARIANT			
0	VT_BSTR		The unit axis pattern is returned.
1	VT_UI2		Work coordinate system selection No.(unit 1) is returned.
2	VT_UI2		Tool coordinate system selection No.(unit 1) is returned.
3	VT_UI2		Unit 1 common status is returned.
4	VT_UI2		Work coordinate system selection No.(unit 2) is returned.
5	VT_UI2		Tool coordinate system selection No.(unit 2) is returned.
6	VT_UI2		Unit 2 common status is returned.
7	VT_ARRAY | VT_I4		
	0	VT_I4	The axis status is returned.
	1	VT_I4	The axis sensor input status is returned.
	2	VT_I4	The error code related to the axis is returned.
	3	VT_I4	The encoder status (Reset it) is returned.
	4	VT_I4	Present location is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("UNIT_AXIS_STATUS2", Array("1:2:3", 1))
```

2.3.2.54. CaoController::Execute("COORDINATE_DATA_RANGE2") command

Acquisition for coordinate system definition data range specification continuousness 2 is executed.

Argument:

VT_ARRAY | VT_UI2		
0	VT_BSTR	The inquiry axis pattern is specified.
1	VT_UI2	The type is specified.
2	VT_UI2	Head coordinate system definition data No. of the inquiry distant future is specified.
3	VT_UI2	The number of inquiry records is specified.

Return value:

VT_ARRAY | VT_VARIANT				
0	VT_ARRAY | VT_I4			
	0	VT_I4	The amount of X coordinates offset is returned.
	1	VT_I4	The amount of Y coordinates offset is returned.
	2	VT_I4	The amount of Z coordinates offset is returned.
	3	VT_I4	The amount of R coordinates offset is returned.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("COORDINATE_DATA_RANGE2", Array("1:2:3", 1, 2, 3))
```

2.3.2.55. CaoController::Execute("MOVE_UNIT_ABSOLUIT ") command

The absolute coordinates specification of unit movement is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	The acceleration is specified.
2	VT_UI2	The deceleration is specified.
3	VT_UI2	The speed is specified.
4	VT_UI2	The positioning operation type is

				specified.
	5	VT_ARRAY | VT_I4		
		n	VT_I4	Coordinate data is absolutely specified.

The return value: It is not.

Usage example:

controller.Execute("MOVE_UNIT_ABSOLUIT", Array("1:2:3", 2, 3, 4, 1, Array(10, 20)))

2.3.2.56. CaoController::Execute("MOVE_UNIT_RELATIVE ") command

The unit relative coordinates specification movement is executed.

Argument:

VT_ARRAY | VT_VARIANT			
0	VT_BSTR		The axis pattern is specified.
1	VT_UI2		The acceleration is specified.
2	VT_UI2		The deceleration is specified.
3	VT_UI2		The speed is specified.
4	VT_UI2		The positioning operation type is specified.
5	VT_ARRAY | VT_I4		
	n	VT_I4	Relative coordinate data is specified.

Return value:

Usage example:

controller.Execute("MOVE_UNIT_RELATIVE", Array("1:2:3", 2, 3, 4, 1, Array(10, 20)))

2.3.2.57. CaoController::Execute("MOVE_UNIT_POSITION_NO") command

The unit position No. specification movement is executed.

Argument:

VT_ARRAY | VT_VARIANT		
0	VT_BSTR	The axis pattern is specified.
1	VT_UI2	The acceleration is specified.
2	VT_UI2	The deceleration is specified.
3	VT_UI2	The speed is specified.
4	VT_UI2	The positioning operation type is specified.
5	VT_UI2	Position No. is specified.

The return value: It is not.

Usage example:

```
Dim vArray As Variant
vArray = controller.Execute("MOVE_UNIT_POSITION_NO", Array("1:2:3", 2, 3, 4, 5, 6))
```

2.3.2.58. CaoController::Execute("MOVE_UNIT_POSITION_NO2") command

Unit position No. specification movement 2 is executed.

Argument:

VT_ARRAY | VT_VARIANT			
	0	VT_BSTR	The axis pattern is specified.
	1	VT_UI2	The acceleration is specified.
	2	VT_UI2	The deceleration is specified.
	3	VT_UI2	The speed is specified.
	4	VT_UI2	The positioning operation type is specified.
	5	VT_UI2	Position No. is specified.

The return value: It is not.

Usage example:

```
controller.Execute("MOVE_UNIT_POSITION_NO2", Array("1:2:3", 2, 3, 4, 1, 5))
```

2.3.3. CaoController::AddVariable method

The read and written data is decided from the connected SEL device by specifying the variable identifier when the Variable object is generated from CaoController.

The specification of AddVariable is shown as follows.

Format

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

When adhering the variable identifier, optional that can be used, please refer to Table2-4 for details.

2.3.4. CaoController::get_VariableNames method

Table2-4It drinks and the variable name list is acquired.

2.3.5. CaoVariable::get_Value property

Data is acquired from the SEL device by the specified option.

2.3.6. CaoVariable::set_Value property

Data is written in the SEL device by the specified option.

2.4. Variable list

2.4.1. Controller class

The followingTable2-4The variable list that can be used with AddVariable of [ni] controller class is described.

Table2-4Controller class variable list

Variable identifier	Explanation	Value		Link
		get	put	
@MAKER_NAME	The manufacturer name is acquired.	✓	-	P.50
@VERSION	The version is acquired.	✓	-	P.50
VERSION_CODE<??>	The version code is acquired.	✓	-	P.50
@POSITION_DATA_NUM	The number of effective position data is acquired.	✓	-	P.51
POSITION_DATA<??>	It is set as the acquisition of the effective position data.	✓	✓	P.51
INPUT_PORT<??>	The input port is acquired.	✓	-	P.52
OUTPUT_PORT<??>	The output port is acquired.	✓	-	P.52
FLAG<??>	The flag is acquired.	✓	-	P.53
INTEGER_VARIABLE<??>	It is set as the acquisition of the integer variable.	✓	✓	P.54
REAL_VARIABLE<??>	It is set as the acquisition of the real number variable.	✓	✓	P.55
STRING_VARIABLE<??>	It is set as the acquisition of the string variable.	✓	✓	P.55
AXIS_STATUS<??>	The axis status is acquired.	✓	-	P.56
PROGRAM_STATUS<??>	The program status is acquired.	✓	-	P.56
@SYSTEM_STATUS	The system status is acquired.	✓	-	P.57
ERROR_INFO<??>	Error details information is acquired.	✓	-	P.58
@EX2_POSITION_DATA_NUM	Number 2 of effective position data is acquired.	✓	-	P.59
EX2_POSITION_DATA<??>	It is set as the acquisition of effective position data 2.	✓	✓	P.59
EX3_POSITION_DATA<??>	It is set as the acquisition of effective position data 3.	✓	✓	P.60
COORDINATE_DATA_RANGE<??>	Coordinate system definition data range continuous specification is acquired.	✓	-	P.61
UNIT_AXIS_STATUS<??>	The unit axis status is acquired.	✓	-	P.62

CHECK_ZONE_DATA_RANGE<??>	Simple interference check zone definition data range continuous specification is acquired.	✓	-	P.63
EX2_UNIT_AXIS_STATUS<??>	Unit axis status 2 is acquired.	✓	-	P.64
EX2_COORDINATE_DATA_RANGE<??>	Coordinate system definition data range specification continuousness 2 is acquired.	✓	-	P.65

- > is an arbitrary character.

2.4.1.1. @MAKER_NAME

The manufacturer name is acquired.

The option: It is not.

Composition of data of return value at get_value:

VT_BSTR	Fixed value "IAI"

2.4.1.2. @VERSION

The version is acquired.

The option: It is not.

Composition of data of return value at get_value:

VT_BSTR	"*. *. *" character string that shows version of present DLL

2.4.1.3. VERSION_CODE<??>

The version code is acquired.

Option:

UNIT_TYPE	The unit type is specified.	0 - 3
DEVICE_NO	Device No. is specified.	0 -

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
0	VT_UI2	Model code
1	VT_UI2	Unit code
2	VT_UI2	Version No.
3	VT_UI2	Time
4	VT_UI2	Time
5	VT_UI2	Time
6	VT_UI2	Time
7	VT_UI2	Time
8	VT_UI2	Time

2.4.1.4. @POSITION_DATA_NUM

The number of effective position data is acquired.

The option: It is not.

Composition of data of return value at get_value:

VT_UI2	Number of effective position data

2.4.1.5. POSITION_DATA<??>

It is set as the acquisition of the effective position data.

Option:

POSITION_NO	First position No. is specified.	0 -
RECORD_NUM	The number of records is specified.	1 -
AXIS_NUM	The number of effective axes is specified.	1 ? 8

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT						
	0	VT_ARRAY | VT_VARIANT				
		0	VT_UI2			Position No.
		1	VT_BSTR			Axis pattern
		2	VT_UI2			Acceleration
		3	VT_UI2			Deceleration
		4	VT_UI2			Speed
		5	VT_ARRAY | VT_I4			
				n	VT_I4	The positional data

Composition of data of set value at put_value:

It is the same as the data composition of the return value of get_value.

2.4.1.6. INPUT_PORT<??>

The input port is acquired.

Option:

PORT_NO	Inquiry beginning port No. is specified.	0 -
PORT_NUM	The number of inquiry ports is specified.	1 -

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
n	VT_UI2	Input port data

2.4.1.7. OUTPUT_PORT<??>

The output port is acquired.

Option:

PORT_NO	Inquiry beginning port No. is specified.	0 -
PORT_NUM	The number of inquiry ports is specified.	1 -

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
n	VT_UI2	Output port data

2.4.1.8. FLAG<??>

The flag is acquired.

Option:

PROGRAM_NO	Program No. is specified.	0 -
START_NO	Beginning variable No. is specified.	0 -
DATA_NUM	The number of flags is specified.	1 -
ARRAY	The array existence is specified. (Return it with VT_UI2 in case of true and the number of of flag =1.)	true / false (The integer variable data is returned with VT_UI2 in case of true and the variable data the number of =1.)

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
n	VT_UI2	Flag data

2.4.1.9. INTEGER_VARIABLE<??>

It is set as the acquisition of the integer variable.

Option:

PROGRAM_NO	Program No. is specified.	0 -
START_NO	Beginning variable No. is specified.	0 -
DATA_NUM	The number of variable data is specified.	1 -
ARRAY	The array existence is specified. (Return it with VT_UI2 in case of true and the number of of flag =1.)	true / false (The integer variable data is returned with VT_UI2 in case of true and the variable data the number of =1.)

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
n	VT_UI2	Integer variable data

Composition of data of set value at put_value

It is the same as the data composition of the return value of get_value.

2.4.1.10. REAL_VARIABLE<??>

It is set as the acquisition of the real number variable.

Option:

PROGRAM_NO	Program No. is specified.	0 -
START_NO	Beginning variable No. is specified.	0 -
DATA_NUM	The number of variable data is specified.	1 -
ARRAY	The array existence is specified. (Return it with VT_UI2 in case of true and the number of of flag =1.)	true / false (The integer variable data is returned with VT_UI2 in case of true and the variable data the number of =1.)

Composition of data of return value at get_value:

VT_ARRAY | VT_R8		
n	VT_R8	Real number variable data

Composition of data of set value at put_value:

It is the same as the data composition of the return value of get_value.

2.4.1.11. STRING_VARIABLE<??>

It is set as the acquisition of the string variable.

Option:

PROGRAM_NO	Program No. is specified.	0 -
START_NO	Beginning variable No. is specified.	0 -
DATA_NUM	The number of variable data is specified.	1 -

Composition of data of return value at get_value:

VT_BSTR	String variable data

Composition of data of set value at put_value:

It is the same as the data composition of the return value of get_value.

2.4.1.12. AXIS_STATUS<??>

The axis status is acquired.

Option:

AXIS_PATTERN	The inquiry axis pattern is specified.	1 - 8

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_I4		
	0	VT_I4	Axis status
	1	VT_I4	Axis sensor input status
	2	VT_I4	Error code related to axis
	3	VT_I4	Encoder status (Reset it).
	4	VT_I4	Present location

2.4.1.13. PROGRAM_STATUS<??>

The program status is acquired.

Option:

PROGRAM_NO	Program No. is specified.	0 -

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2		
0	VT_UI2	Status
1	VT_UI2	Program step No. when being executing it
2	VT_UI2	Program dependence error code
3	VT_UI2	Error generation step No.

2.4.1.14. @SYSTEM_STATUS

The system status is acquired.

The option: It is not.

Composition of data of return value at get_value:

VT_ARRAY | VT_UI2

0	VT_UI2	System mode
1	VT_UI2	Heavy level system error No.
2	VT_UI2	Latest system error No.
3	VT_UI2	System status ..byte.. one
4	VT_UI2	System status ..byte.. two
5	VT_UI2	System status ..byte.. three
6	VT_UI2	System status ..byte.. four

2.4.1.15. ERROR_INFO<??>

Error details information is acquired.

Option:

TYPE1	Type 1 is specified.	0 -
TYPE2	Type 2 is specified.	0 -
RECORD_NO	Record No. is specified.	0 -

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT		
0	VT_UI4	Error No.
1	VT_UI4	Detailed information 1
2	VT_UI4	Detailed information 2
3	VT_UI4	Detailed information 3
4	VT_I4	Detailed information 4
5	VT_UI4	Detailed information 5
6	VT_UI4	Detailed information 6
7	VT_UI4	Detailed information 7
8	VT_UI4	Detailed information 8
9	VT_BSTR	Message character string

2.4.1.16. @EX2_POSITION_DATA_NUM

Number 2 of effective position data is acquired.

The option: It is not.

Composition of data of return value at get_value:

VT_UI2	Number of effective position data

2.4.1.17. EX2_POSITION_DATA<??>

It is set as the acquisition of effective position data 2.

Option:

POSITION_NO	First position No. is specified.	0 -
RECORD_NUM	The number of records is specified.	1 -
AXIS_NUM	The number of effective axes is specified.	1 - 8

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_VARIANT		
	0	VT_UI2	Position No.
	1	VT_UI2	Axis pattern
	2	VT_UI2	Acceleration
	3	VT_UI2	Deceleration
	4	VT_UI2	Speed
	5	VT_ARRAY | VT_I4	
		n VT_I4	The positional data

Composition of data of set value at put_value

It is the same as the data composition of the return value of get_value.

2.4.1.18. EX3_POSITION_DATA<??>

It is set as the acquisition of effective position data 3.

Option:

OUTPUT_OPERATION	The record format extension specification type is specified.	0 ? 1 0= invalidity 1= effective
POSITION_NO	First position No. is specified.	0 -
RECORD_NUM	The number of records is specified.	1 -
AXIS_NUM	The number of effective axes is specified.	1 - 8

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_VARIANT		
	0	VT_UI2	Position No.
	1	VT_UI2	Axis pattern
	2	VT_UI2	Acceleration
	3	VT_UI2	Deceleration
	4	VT_UI2	Speed
	5	VT_ARRAY | VT_I4	
		n VT_I4	The positional data
	6	VT_UI2	Enhancing data
	7	VT_UI4	Output function code
	8	VT_UI4	Output port flag No.
	9	VT_UI4	Function parameter 1
	10	VT_UI4	Function parameter 2

Composition of data of set value at put_value:

It is the same as the data composition of the return value of get_value.

2.4.1.19. COORDINATE_DATA_RANGE<??>

Coordinate system definition data range continuous specification is acquired.

Option:

TYPE	The type is specified.	0 ? 1 0= work coordinate system definition data 1= tool coordinate system definition data
DATA_NO	Head coordinate system definition data No. of the inquiry distant future is specified.	0 -
RECORD_NUM	The number of inquiry records is specified.	1 -

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT				
0	VT_ARRAY | VT_I4			
		0	VT_I4	Amount of X coordinates offset
		1	VT_I4	Amount of Y coordinates offset
		2	VT_I4	Amount of Z coordinates offset
		3	VT_I4	Amount of R coordinates offset

2.4.1.20. UNIT_AXIS_STATUS<??>

The unit axis status is acquired.

Option:

AXIS_PATTERN	The inquiry axis pattern is specified.	1 - 8
TYPE	The type is specified.	0 -

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT			
0	VT_UI2		Work coordinate system selection No.
1	VT_UI2		Tool coordinate system selection No.
2	VT_UI2		Axis common status
3	VT_ARRAY | VT_I4		
	0	VT_I4	Axis status
	1	VT_I4	Axis sensor input status
	2	VT_I4	Error code related to axis
	3	VT_I4	Encoder status (Reset it).
	4	VT_I4	Present location

2.4.1.21. CHECK_ZONE_DATA_RANGE<??>

Simple interference check zone definition data range continuous specification is acquired.

Option:

DATA_NO	Inquiry head and simplicity interference check zone definition data No. is specified.	0 -
RECORD_NUM	The number of inquiry records is specified.	1 -
AXIS_NUM	The number of axes is specified.	1 - 4

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT				
0	VT_ARRAY | VT_VARIANT			
	0	VT_UI4		Simple interference check zone definition coordinates effective axis pattern
	1	VT_ARRAY | VT_I4		
		n	VT_I4	Simple interference check zone definition coordinates 1
	2	VT_ARRAY | VT_I4		
		n	VT_I4	Simple interference check zone definition coordinates 2
	3	VT_UI2		Outputting physical output port No. or global flag No. when invading
	4	VT_UI2		Error type specification when invading

2.4.1.22. EX2_UNIT_AXIS_STATUS<??>

Unit axis status 2 is acquired.

Option:

AXIS_PATTERN	The inquiry axis pattern is specified.	1 - 8
TYPE	The type is specified.	0 -

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT				
0	VT_UI2			Work coordinate system selection No.
1	VT_UI2			Tool coordinate system selection No.
2	VT_UI2			Axis common status
3	VT_ARRAY | VT_I4			
	0	VT_I4		Axis status
	1	VT_I4		Axis sensor input status
	2	VT_I4		Error code related to axis
	3	VT_I4		Encoder status (Reset it).
	4	VT_I4		Present location

2.4.1.23. EX2_COORDINATE_DATA_RANGE<??>

Coordinate system definition data range specification continuousness 2 is acquired.

Option:

AXIS_PATTERN	The inquiry axis pattern is specified.	1 -
TYPE	The type is specified.	0 - 1 0= work coordinate system definition data 1= tool coordinate system definition data
DATA_NO	Head coordinate system definition data No. of the inquiry distant future is specified.	0 -
RECORD_NUM	The number of inquiry records is specified.	1 -

Composition of data of return value at get_value:

VT_ARRAY | VT_VARIANT

0	VT_BSTR	Unit axis pattern
1	VT_UI2	Work coordinate system selection No.(unit 1)
2	VT_UI2	Tool coordinate system selection No.(unit 1)
3	VT_UI2	Unit 1 common status
4	VT_UI2	Work coordinate system selection No.(unit 2)
5	VT_UI2	Tool coordinate system selection No.(unit 2)
6	VT_UI2	Unit 2 common status
7	VT_ARRAY | VT_I4	

0	VT_I4	Axis status
1	VT_I4	Axis sensor input status
2	VT_I4	Error code related to axis

			3	VT_I4	Encoder status (Reset it).
			4	VT_I4	Present location

2.5. Error code

In this provider, the following and original the error code exists. (Table2-5Reference)
About the ORiN2 commonness error, "ORiN2 プログラミングガイド Please refer to the chapter of the error code of".

Table2-5Original error code table

Error name	Error number	Explanation
E_AXIS_DUPLICATION	0x80110001	There is repetition in the axis number specification.
E_OUT_OF_DATA_RANGE	0x80110002	The one outside the value range exists in the specified data.
E_PACKET_CHECKSUM	0x80110003	An abnormal packet by the communication was generated.
E_MEMORY_AREA_ALLOCATION	0x80110004	An internal memory partitioning failed.
--	0x80100***	The error occurred by the device. ***It is an error code of 3 digit. peel off Please refer to the error code of the manual of the target model.

3. Sample program

The easy sample written in the SEL device is shown as follows.

Precondition:
·Target the PAC script of RC8.
·The COM port of the SEL device is assumed to be "1".

List 3-1 Sample.pcs

```
Sub Main

        ..'.. object
        Dim caoCtrl as Object
        Dim caoVal as Object

        Controller object ..'.. making
        caoCtrl = cao.AddController("SEL", "CaoProv.IAI.SEL", "",
"conn=com:1,station=99,timeout=500")

        ..'.. servo ON
        caoCtrl.Execute "SERVO", Array("1", 1)

        Starting point ..'.. return
        caoCtrl.Execute "ORIGIN", Array("1", 100, 100)
        It ..'.. waits until the starting point return is completed.
        delay 10000

        Absolute value ..'.. movement
        caoCtrl.Execute "MOVE_ABSOLUITE", Array("1", 100, 100 ,100 , Array(300000))
        It ..'.. waits until the absolute value movement is completed.
        delay 10000

End Sub
```