ETAC
NEO series (command port) provider

Version 1.0.0

User's guide

September 3, 2018

Remarks:

This document is translated into English by machine translation.

【 revision history 】

Version	Date	Content
1.0.0	2018-9-3	First edition.

【 hardware 】

Model	Version	Notes

Contents

1. Introduction

This book is an user's guide of the ETAC NEO COM provider that reads data to low temperature constant temperature [**] [shimeutsuwa] and the low temperature constant temperature machine (Hereafter, it is named the NEO series) made of the [etakku] division of Kusumoto transformation Ltd..

It becomes easy to acquire data from the NEO series by using the ETAC NEO COM provider.

This book explains the function of the ETAC NEO COM provider and the mounting method.

2. Outline of provider

2.1. Outline

The ETAC NEO COM provider is CAO provider that reads data by communicating by the Ethernet/IP(TCP) connection for Ruby I/F of the NEO series.

In this provider, acquire information by using a command above-mentioned Ruby I/F port (The port number: 10052).

The file format is DLL(Dynamic Link Library), and when using it from the CAO engine, it is dynamically loaded.

Is installed ORiN2SDK when you use the ETAC NEO COM provider?Table2-1It is necessary to register the registry referring to [wo].

Table2-1ETAC NEO COM provider

File name	CaoProvEtacNeoCom.dll
ProgID	CaoProv.ETAC.NEO_COM
Registry registration	regsvr32 CaoProvEtacNeoCom.dll
Blotting out of registry registration	regsvr32 /u CaoProvEtacNeoCom.dll

2.2. Method property

2.2.1. CaoWorkspace::AddController method

The ETAC NEO COM provider connects the communication referring to connected parameter for the communication at AddController.

Format AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,<bstrPCName:BSTR>, <bstrOption:BSTR>)

bstrCtrlName	:	[in]	Controller name
bstrProvName	:	[in]	Provider name Fixed value = "CaoProv.ETAC.NEO_COM"
bstrPCName	:	[in]	Execution machine name of provider
bstrOption	:	[in]	Optional character string

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

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

Option	Explanation
Conn=< connected parameter >	Indispensability. Communication form and connected parameter. (reference 2.2.1.1)
ConnTimeout=<Connected time-out time >	Specify the timeout period (millisecond) when it connects it. Range of value: 0 4294967295- (4294967295: Infinite waiting) Default: 500
Timeout =< timeout period >	Specify the timeout period (millisecond). Range of value: 0 ? 4294967295 (4294967295: Infinite waiting) Default: 3000

2.2.1.1. Conn is optional.

Connected parameter character string of optional Conn is shown as follows. A possible omission is shown here in the square bracket (""). Moreover, the underlined part under the explanation of each parameter becomes a default value when the option is not specified.

Ethenet device

"Conn=ETH:<IP Address>[:<Port No>]"

<IP Address> : TCP/IP Internet Protocol address.
Example:"127.0.0.1","192.168.0.1"

<Port No> : TCP/IP connection port number.
10052(command port) and 10053(inquiry port)
It is possible to specify it voluntarily.
- The variable and the command that can be used with the port are changed.
Please refer to chapter 2.3 and Table 3-1 for details.

Usage
example

```
Dim caoCtrl As CaoController
Set caoCtrl = caoWorkspace.AddController("EtacNeoCom", _
    "CaoProv.ETAC.NEO_COM", "", _
    "Conn=ETH:192.168.0.1:10052,ConnTimeout=500,Timeout=3000")
```

2.2.2. CaoController::AddVariable method

The AddVariable method of the CaoController class is a method for making the variable object to read data to the NEO series.

Please refer to 2.3 variable list for the variable identifier that can be specified.

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

bstrVariableName	:	[in]	Variable identifier
bstrOption	:	[in]	Optional character string

There is no variable that can specify an optional character string in the ETAC NEO COM provider.

Usage

example

```
Dim caoVar As CaoVariable
Set caoVar = caoCtrl.AddVariable("@TEST_INFO", "")
```

2.2.3. CaoController::get_VariableNames property

The VariableNames property of the CaoController class acquires the list of the system variable that can be specified by the AddVariable method.

Please refer to 2.3 variable list for the variable identifier that can be specified.

2.2.4. CaoController::Execute method

The Execute method of the CaoController class is a method for the execution of the command.

Please refer to three command reference for details of each command.

Format Execute(<bstrCommandName:VT_BSTR>,[<vntParam:VT_VARIANT>])

bstrCommandName	:	[in]	Command name
vntParam	:	[in]	Parameter

Usage

example

```
Dim vTestInfo As Variant
vTestInfo = caoCtrl.Execute("GetTestInfo")
```

2.2.5. CaoVariable::get_Value

The Value property of the CaoVariable class acquires the value of the variable that corresponds to the object.

Please refer to 2.3 variable list for details of each variable.

2.2.6. CaoVariable::put_ID

Only @PROGRAM_DATA becomes effective.

Set ID of the variable.

This ID becomes step No. at the professional ram data acquisition.

About details2.3.1.25@PROGRAM_DATARefer to [wo].

2.2.7. CaoVariable::get_ID

Only @PROGRAM_DATA becomes effective.

Acquire ID of the variable set now.

2.3. Variable list

2.3.1. Controller class

The variable list that can be used in the CaoController class is shown below.

Variable identifier	Data type	Explanation	Port that can be used		Attribute	
			Command	Inquiry	get	put
@MAKER_NAME	VT_BSTR	Return the manufacturer name.	−	−	−	−
@VERSION	VT_BSTR	Return version information on provider DLL.	−	−	−	−
@DEVICE_INFO	VT_ARRAY | VT_BSTR	Return equipment information on the NEO series.	−	−	−	−
@TEMP	VT_R8	Return the temperature of the NEO series.	−	−	−	−
@HUMI	VT_I4	Return the humidity of the NEO series.	−	−	−	−
@MODE	VT_ARRAY | VT_VARIANT	Return the state of the NEO series.	−	−	−	−
@ALARM	VT_I4	Return the error of the NEO series.	−	−	−	−
@DATE	VT_ARRAY | VT_VARIANT	Return the date of the NEO series.	−	−	−	−
@OPERATING_CONDITION	VT_ARRAY | VT_VARIANT	Return driving the NEO series.	−	×	−	−
@STEP_INFO	VT_ARRAY | VT_VARIANT	Return step information on the NEO series.	−	×	−	−
@TEST_INFO	VT_ARRAY | VT_VARIANT	Return examination information on the NEO series.	−	×	−	−
@DATE_TIME	VT_ARRAY | VT_UI2	Return the date of the NEO series.	−	×	−	−
@CURRENT_OPERATING_INFO	VT_ARRAY | VT_VARIANT	Return operating information on the NEO series now.	−	×	−	−
@DRY_BULB_CORRECTION	VT_R8	Return the [kawakyuu] correction of the NEO series.	−	×	−	−

@WET_BULB_CORRECTION	VT_R8	[Kyuu] correction of the NEO series ..damp.. is returned.	−	×	−	−
@HUMIDIFIER_WATER_SUPPLY_OPTION	VT_ARRAY | VT_I4	Return a humidifying water optional water supply of the NEO series.	−	×	−	−
@IP_ADDRESS	VT_ARRAY | VT_I4	Return Internet Protocol address of the NEO series.	−	×	−	−
@WINDOW_LAMP_SETTING	VT_I4 /VT_BSTR	Return the window lamp setting of the NEO series.	−	×	−	−
@UPPER_LOWER_LIMIT_WARNING_MODE	VT_I4	Return the bound pair warning mode of the NEO series.	−	×	−	−
@BUZZER_SOUND_PROCESS	VT_I4	Return the buzzer sound processing of the NEO series.	−	×	−	−
@TEMPERATURE_SETTING_RANGE	VT_ARRAY | VT_R8	Return the range of the temperature setting of the NEO series.	−	×	−	−
@HUMIDITY_SETTING_RANGE	VT_ARRAY | VT_R8	Return the range of the humidity setting of the NEO series.	−	×	−	−
@HUMIDITY_CONTROLLED_TEMPERATURE_RANGE	VT_ARRAY | VT_R8	Return the range of the NEO series of the humidity control temperature.	−	×	−	−
@UPPER_LOWER_LIMIT_TEMPERATURE_HUMIDITY	VT_ARRAY | VT_I4	Return the bound pair temperature humidity of the NEO series.	−	×	−	−
@PROGRAM_DATA	VT_ARRAY | VT_VARIANT	Return the program data of the NEO series.	−	×	−	−
@REPEAT_DATA	VT_ARRAY | VT_VARIANT	Return the repetition data of the NEO series.	−	×	−	−
@HUMIDIFICATION_DELAY	VT_I4	Return the humidifying delay of the NEO series.	−	×	−	−
@TEST_END_OPERATION	VT_I4	Return the examination end driving of the NEO series.	−	×	−	−
@AUTO_START_TIME	VT_ARRAY | VT_UI2	Return the auto start time of the NEO series.	−	×	−	−
@START_STEP	VT_I4	Return the start step of the NEO series.	−	×	−	−

@AUTO_STOP_REMAINING_TIME	VT_ARRAY | VT_UI2	Return the auto stop remainder time of the NEO series.	-	×	-	-

Table2-3Controller class system variable list

2.3.1.1. @MAKER_NAME

Return the manufacturer name.

VT_BSTR	Manufacturer name

2.3.1.2. @VERSION

Return version information on provider DLL.

VT_BSTR	Version information on provider DLL

2.3.1.3. @DEVICE_INFO

Return equipment information on the NEO series.

VT_ARRAY | VT_BSTR		
	0	ETAC Engineering Co. Ltd.
	1	Constant temperature tank model
	1	Constant temperature tank Lot No.
	2	Library version

2.3.1.4. @TEMP

Return the temperature of the NEO series.

VT_R8	Temperature of testing room

2.3.1.5. @HUMI

Return the humidity of the NEO series.

VT_I4	Humidity of testing room

2.3.1.6. @MODE

Return the state of the NEO series.

VT_ARRAY | VT_VARIANT				
0	VT_BSTR		TEST: be examining it/ END: It is non-examining it.	
1	VT_ARRAY | VT_UI2		Expected completion	
		0	Age	
		1	The moon	
		2	Day	
2	VT_ARRAY | VT_UI2		Expected completion	
		0	Time	
		1	Amount	
		2	Second	
3	VT_BSTR		ALARM	There is trouble or alarm.
			RUN	Driving
			STOP	Stop
			HOLD	It is holding.
			RUN-UNREACHED	Driving (untrodden)
			END	End
			WAIT-AUTOSTART	Auto start standby
			PAUSE	Temporary stop
			CONTINUE	Examination end driving
			DELAYED-HUMIDIFICATION	Humidifying delay
			DEFROST	[Defurosuto]
			NONFROST	[Nonfurosuto]

2.3.1.7. @ALARM

Return the error of the NEO series.

VT_I4	0	:	Trouble or alarm none
	101～199	:	Alarm
	201～299	:	Trouble

2.3.1.8. @DATE

Return the date of the NEO series.

VT_ARRAY | VT_VARIANT		
0	VT_ARRAY | VT_UI2	Date now

			0	Age
			1	The moon
			2	Day
	1	VT_ARRAY | VT_UI2		Time now
			0	Time
			1	Amount
			2	Second

2.3.1.9. @OPERATING_CONDITION

Return driving the NEO series.

VT_ARRAY | VT_VARIANT		
0	VT_R8	Temperature
1	VT_R8	Humidity
2	VT_I4	Number of steps
3	VT_I4	Error code
4	VT_I4	Mode
5	VT_I4	Partial repetition passage frequency
6	VT_BSTR	Mode of operation F: Fixed value and P: Program
7	VT_BTSR	Weight setting W : ON, R : OFF
8	VT_I4	0 fixation
9	VT_ARRAY | VT_UI2	Expected completion time

0	Age
1	The moon
2	Day
3	Time
4	Amount
5	Second

2.3.1.10. @STEP_INFO

Return step information on the NEO series.

VT_ARRAY | VT_VARIANT		
0	VT_I4	Present step

1	VT_ARRAY | VT_UI2		Step elapsed time
		0	Time
		1	Amount
2	VT_ARRAY | VT_UI2		Step setting time
		0	Time
		1	Amount

2.3.1.11. @TEST_INFO

Return examination information on the NEO series.

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_UI2		Time now
		0	Age
		1	The moon
		2	Day
		3	Time
		4	Amount
		5	Second
1	VT_R8		Present preset temperature
2	VT_R8		Present temperature
3	VT_R8/VT_BSTR		Present set humidity When the temperature is driven, "--.."... fixation
4	VT_R8		Present humidity
5	VT_R8		Dry bulb temperature
6	VT_R8		Wet bulb temperature
7	VT_R8		Relative humidity
8	VT_R8		Upper bound temperature
9	VT_R8		Lower bound temperature
10	VT_R8		Upper bound humidity
11	VT_R8		Lower bound humidity
12	VT_ARRAY | VT_UI2		Set time now
		0	Time
		1	Amount
13	VT_ARRAY | VT_UI2		Local elapsed time
		0	Time
		1	Amount

14	VT_ARRAY | VT_UI2		Expected completion time
		0	Age
		1	The moon
		2	Day
		3	Time
		4	Amount
		5	Second
15	VT_I4		Step
16	VT_ARRAY | VT_UI2		Step setting time
		0	Time
		1	Amount
17	VT_ARRAY | VT_UI2		Step elapsed time
		0	Time
		1	Amount
18	VT_I4		Partial repetition passage frequency
19	VT_I4		State of equipment
20	VT_I4		Error code
21	VT_I4		Panel/web distinction 0 : Panel constant (?) 1 : Web 2 : Unused 3 : Panel program 1 4 : Panel program 2 5 : Panel program 3 6 : Panel program 4 7 : Panel program 5 8 : Panel program 6 9 : Panel program 7 A : Panel program 8 B : Panel program 9
22	VT_I4		Stop distinction 0= web halt condition/1= panel halt condition
23	VT_I4		Driving beginning distinction New 0= drive/1= driving restart

2.3.1.12. @DATE_TIME

Return the date of the NEO series.

VT_ARRAY | VT_UI2	
0	Age
1	The moon
2	Day
3	Time
4	Amount
5	Second

2.3.1.13. @CURRENT_OPERATING_INFO

Return operating information on the NEO series now.

VT_ARRAY | VT_VARIANT			
0	VT_ARRAY | VT_UI2		Time now
	0	Age	
	1	The moon	
	2	Day	
	3	Time	
	4	Amount	
	5	Second	
1	VT_I4		Present step
2	VT_ARAAY | VT_UI2		Step time now
	0	Time	
	1	Amount	
3	VT_I4		Whole repetition frequency of present
4	VT_I4		Mode
5	VT_I4		Error code
6	VT_R8		Ambient temperature
7	VT_R8		Dry bulb temperature
8	VT_R8		Wet bulb temperature
9	VT_R8		Relative humidity
10	VT_R8		One in evaporation temperature
11	VT_R8		Absorption temperature degree 1
12	VT_R8		One in exhalation temperature

	13	VT_R8	2.3 in evaporation temperature
	14	VT_R8	Absorption temperature degree 2.3
	15	VT_R8	2.3 in exhalation temperature
	16	VT_R8	Three in condensation temperature
	17	VT_R8	Specimen temperature
	18	VT_I4	Frequency of inverter
	19	VT_I4	Warming heater output
	20	VT_I4	Humidifying heater output
	21	VT_I4	Cooling output
	22	VT_BSTR	Digital input/output (24 characters)

2.3.1.14. @DRY_BULB_CORRECTION

Return the [kawakyuu] correction of the NEO series.

VT_R8	[Kawakyuu] correction value

2.3.1.15. @WET_BULB_CORRECTION

[Kyuu] correction of the NEO series ..damp.. is returned.

VT_R8	[Kyuu] correction ..damp.. value

2.3.1.16. @HUMIDIFIER_WATER_SUPPLY_OPTION

Return a humidifying water optional water supply of the NEO series.

VT_ARRAY | VT_I4		
0	The water supply is optional.	0= zilch 1= having
1	Feed mode	0= zilch 1= having
2	The humidifying water is optional.	0= zilch 1= having
3	The pure water machine is optional.	0= zilch 1= having

2.3.1.17. @IP_ADDRESS

Return Internet Protocol address of the NEO series.

VT_ARRAY | VT_I4	
0	The first byte of IPV4 address
1	The second byte of IPV4 address
2	The third byte of IPV4 address
3	The fourth byte of IPV4 address

2.3.1.18. @WINDOW_LAMP_SETTING

Return the window lamp setting of the NEO series.

VT_I4/VT_BSTR	Automatic turning off time/ENDLESS

2.3.1.19. @UPPER_LOWER_LIMIT_WARNING_MODE

Return the bound pair warning mode of the NEO series.

VT_I4	0= alarm treatment 1= treatment of trouble

2.3.1.20. @BUZZER_SOUND_PROCESS

Return the buzzer sound processing of the NEO series.

VT_I4	It 1 that 0= rings= doesn't ring.

2.3.1.21. @TEMPERATURE_SETTING_RANGE

Return the range of the temperature setting of the NEO series.

VT_ARRAY | VT_R8		
0	Range LOW side	-99.9～200.0 ℃
1	Range HIGH side	-99.9～200.0 ℃

2.3.1.22. @HUMIDITY_SETTING_RANGE

Return the range of the humidity setting of the NEO series.

VT_ARRAY | VT_R8		
0	Range LOW side	0.0～100.0 %RH
1	Range HIGH side	0.0～100.0 %RH

2.3.1.23. @HUMIDITY_CONTROLLED_TEMPERATURE_RANGE

Return the range of the NEO series of the humidity control temperature.

VT_ARRAY | VT_R8			
	0	Range LOW side	0.0～ 99.9 ℃
	1	Range HIGH side	0.0～ 99.9 ℃

2.3.1.24. @UPPER_LOWER_LIMIT_TEMPERATURE_HUMIDITY

Return the bound pair temperature humidity of the NEO series.

VT_ARRAY | VT_I4		
	0	Upper bound temperature
	1	Lower bound temperature
	2	Upper bound humidity
	3	Lower bound humidity

2.3.1.25. @PROGRAM_DATA

Return the program data of the NEO series.

Acquire the program data of step No. specified with put_ID.

VT_ARRAY | VT_VARIANT				
	0	VT_I4	Step No.000~049	
	1	VT_R8/VT_BSTR	-99.9~200.0 in temperature ℃/BLANK	
	2	VT_I4	0~100 in humidity %RH	
	3	VT_ARRAY | VT_UI2	Step time	
		0	Time	
		1	Amount	
	4	VT_I4	Weight	0:OFF 10:Temperature WAIT 1:Humidity WAIT 11:WAIT at each [nurushime]
	5	VT_I4	Time signal	0:OFF 10:TS1-ON 1:TS2-ON 11:TS1-ON/TS2-ON

6	VT_I4	Freezer	0:AUTO 1:OFF 2:ON 3:Ref1ON 4:Ref2ON 6:Ref3ON 7:Ref1+Ref2ON
7	VT_I4	0 fixation	
8	VT_I4	[Defurosuto]	0:OFF 1:ON

2.3.1.26. @REPEAT_DATA

Return the repetition data of the NEO series.

VT_ARRAY | VT_VARIANT			
0	VT_I4/VT_BSTR	Whole repetition frequency	0～999/ENDLESS
1	VT_I4	Partial repetition start step	000～049
2	VT_I4	Partial repetition end step	000～049
3	VT_I4	Partial repetition frequency	000～049

2.3.1.27. @HUMIDIFICATION_DELAY

Return the humidifying delay of the NEO series.

VT_I4	Humidifying delay	0:OFF 1:ON

2.3.1.28. @TEST_END_OPERATION

Return the examination end driving of the NEO series.

VT_I4	Examination end driving	0:OFF 1:ON

2.3.1.29. @AUTO_START_TIME

Return the auto start time of the NEO series.

VT_ARRAY | VT_UI2	
0	Age
1	The moon
2	Day

	3	Time
	4	Amount
	5	Second

2.3.1.30. @START_STEP

Return the start step of the NEO series.

VT_I4	Start step

2.3.1.31. @AUTO_STOP_REMAINING_TIME

Return the auto stop remainder time of the NEO series.

VT_ARRAY | VT_UI2		
	0	Time
	1	Amount

3. Command reference

3.1. Controller class

Table3-1CaoController::Execute command list

Command	Function	Port that can be used		Page
		Command	Inquiry	
GetDeviceInfo	Acquire equipment information on the NEO series.	−	−	26
GetTemp	Acquire the temperature of the NEO series.	−	−	27
GetHumi	Acquire the humidity of the NEO series.	−	−	27
GetMode	Acquire the state of the NEO series.	−	−	27
GetAlarm	Acquire the error of the NEO series.	−	−	28
GetDate	Acquire the date of the NEO series.	−	−	28
GetOperatingCondition	Acquire driving the NEO series.	−	×	29
GetStepInfo	Acquire step information on the NEO series.	−	×	29
GetTestInfo	Acquire examination information on the NEO series.	−	×	30
GetDateTime	Acquire the date of the NEO series.	−	×	30
GetCurrentOperatingInfo	Acquire operating information on the NEO series now.	−	×	30
GetDryBulbCorrection	Acquire the [kawakyuu] correction of the NEO series.	−	×	31
GetWetBulbCorrection	[Kyuu] correction of the NEO series ..damp.. is acquired.	−	×	31
GetHumidifierWaterSupplyOption	Acquire a humidifying water optional water supply of the NEO series.	−	×	32
GetIPAddress	Acquire Internet Protocol address of the NEO series.	−	×	32
GetWindowLampSetting	Acquire the window lamp setting of the NEO series.	−	×	33
GetUpperLowerLimitWarningMode	Acquire the bound pair warning mode of the NEO series.	−	×	33

GetBuzzerSoundProcess	Acquire the buzzer sound processing of the NEO series.	−	×	33
GetTemperatureSettingRange	Acquire the range of the temperature setting of the NEO series.	−	×	34
GetHumiditySettingRange	Acquire the range of the humidity setting of the NEO series.	−	×	34
GetHumidityControlledTemperatureRange	Acquire the range of the NEO series of the humidity control temperature.	−	×	35
GetUpperLowerLimitTemperatureHumidity	Acquire the bound pair temperature humidity of the NEO series.	−	×	35
GetProgramData	Acquire the program data of the NEO series.	−	×	36
GetRepeatData	Acquire the repetition data of the NEO series.	−	×	36
GetHumidificationDelay	Acquire the humidifying delay of the NEO series.	−	×	37
GetTestEndOperation	Acquire the examination end driving of the NEO series.	−	×	37
GetAutoStartTime	Acquire the auto start time of the NEO series.	−	×	37
GetStartStep	Acquire the start step of the NEO series.	−	×	38
GetAutoStopRemainingTime	Acquire the auto stop remainder time of the NEO series.	−	×	38

3.1.1. CaoController::Execute("GetDeviceInfo") command

Acquire equipment information on the NEO series.

Format GetDeviceInfo

Argument : None
Return value : Equipment information (VT_ARRAY | VT_BSTR)
@DEVICE_INFOReference

Usage

example

```
Dim vDeviceInfo As Variant
vDeviceInfo = caoCtrl.Execute("GetDeviceInfo")
```

3.1.2. CaoController::Execute("GetTemp") command

Acquire the temperature of the NEO series.

Format GetTemp

Argument : None
Return value : Temperature (VT_R8)
@TEMPReference

Usage
example

```
Dim vTemp As Variant
vTemp = caoCtrl.Execute("GetTemp")
```

3.1.3. CaoController::Execute("GetHumi") command

Acquire the humidity of the NEO series.

Format GetHumi

Argument : None
Return value : Humidity (VT_I4)
@HUMIReference

Usage
example

```
Dim vHumi As Variant
vHumi = caoCtrl.Execute("GetHumi")
```

3.1.4. CaoController::Execute("GetMode") command

Acquire the state of the NEO series.

Format GetMode

Argument : None
Return value : State (VT_ARRAY | VT_VARIANT)
@MODEReference

Usage
example

```
Dim vMode As Variant
vMode = caoCtrl.Execute("GetMode")
```

3.1.5. CaoController::Execute("GetAlarm") command

Acquire the error of the NEO series.

Format GetAlarm

Argument : None
Return value : Error (VT_I4)
@ALARMReference

Usage
example

```
Dim vAlarm As Variant
vAlarm = caoCtrl.Execute("GetAlarm")
```

3.1.6. CaoController::Execute("GetDate") command

Acquire the date of the NEO series.

Format GetDate

Argument : None
Return value : Date (VT_ARRAY | VT_VARIANT)
@DATEReference

Usage

example

```
Dim vDate As Variant
vDate = caoCtrl.Execute("GetDate")
```

3.1.7. CaoController::Execute("GetOparatingCondition") command

Acquire driving the NEO series.

Format GetOperatingCondition

Argument : None
Return value : Examination information (VT_ARRAY | VT_VARIANT)
@OPERATING_CONDITIONReference

Usage

example

```
Dim vOperatingCondition As Variant
vOperatingCondition = caoCtrl.Execute("GetOperatingCondition")
```

3.1.8. CaoController::Execute("GetStepInfo") command

Acquire step information on the NEO series.

Format GetStepInfo

Argument : None
Return value : Step information (VT_ARRAY | VT_VARIANT)
@STEP_INFOReference

Usage

example

```
Dim vStepInfo As Variant
vStepInfo = caoCtrl.Execute("GetStepInfo")
```

3.1.9. CaoController::Execute("GetTestInfo") command

Acquire examination information on the NEO series.

Format GetTestInfo

Argument : None
Return value : Examination information (VT_ARRAY | VT_VARIANT)
@TEST_INFOReference

Usage example

```
Dim vTestInfo As Variant
vTestInfo = caoCtrl.Execute("GetTestInfo")
```

3.1.10. CaoController::Execute("GetDateTime") command

Acquire the date of the NEO series.

Format GetDateTime

Argument : None
Return value : Date (VT_ARRAY | VT_UI2)
@DATE_TIMEReference

Usage example

```
Dim vDateTime As Variant
vDateTime = caoCtrl.Execute("GetDateTime")
```

3.1.11. CaoController::Execute("GetCurrentOperatingInfo") command

Acquire operating information on the NEO series now.

Format GetCurrentOperatingInfo

Argument : None
Return value : Present operating information (VT_ARRAY | VT_VARIANT)
@CURRENT_OPERATING_INFOReference

Usage
example

```
Dim vCurrentOperatingInfo As Variant
vCurrentOperatingInfo = caoCtrl.Execute("GetCurrentOperatingInfo")
```

3.1.12. CaoController::Execute("GetDryBulbCorrection") command

Acquire the [kawakyuu] correction of the NEO series.

Format GetDryBulbCorrection

Argument : None
Return value : [Kawakyuu] correction (VT_R8)
@DRY_BULB_CORRECTIONReference

Usage
example

```
Dim vDryBulbCorrection As Variant
vDryBulbCorrection = caoCtrl.Execute("GetDryBulbCorrection")
```

3.1.13. CaoController::Execute("GetWetBulbCorrection") command

[Kyuu] correction of the NEO series ..damp.. is acquired.

Format GetWetBulbCorrection

Argument : None
Return value : [Kyuu] ..damp.. correction (VT_R8)
@WET_BULB_CORRECTIONReference

Usage

example

```
Dim vWetBulbCorrection As Variant
vWetBulbCorrection = caoCtrl.Execute("GetWetBulbCorrection")
```

3.1.14. CaoController::Execute("GetHumidifierWaterSupplyOption") command

Acquire a humidifying water optional water supply of the NEO series.

Format GetHumidifierWaterSupplyOption

Argument	:	None
Return value	:	Option (VT_ARRAY | VT_I4) of humidifying water water supply @HUMIDIFIER_WATER_SUPPLY_OPTIONReference

Usage

example

```
Dim vHumidifierWaterSupplyOption As Variant
vHumidifierWaterSupplyOption = caoCtrl.Execute("GetHumidifierWaterSupplyOption")
```

3.1.15. CaoController::Execute("GetIPAddress") command

Acquire Internet Protocol address of the NEO series.

Format GetIPAddress

Argument	:	None
Return value	:	Internet Protocol address (VT_ARRAY | VT_I4) @IP_ADDRESSReference

Usage

example

```
Dim vIPAddress As Variant
```

```
vIPAddress = caoCtrl.Execute("GetIPAddress")
```

3.1.16. CaoController::Execute("GetWindowLampSetting") command

Acquire the window lamp setting of the NEO series.

Format GetWindowLampSetting

Argument : None
Return value : Window lamp setting (VT_I4/VT_BSTR)
@WINDOW_LAMP_SETTINGReference

Usage example

```
Dim vWindowLampSetting As Variant
vWindowLampSetting = caoCtrl.Execute("GetWindowLampSetting")
```

3.1.17. CaoController::Execute("GetUpperLowerLimitWarningMode") command

Acquire the bound pair warning mode of the NEO series.

Format GetUpperLowerLimitWarningMode

Argument : None
Return value : Bound pair warning mode (VT_I4)
@UPPER_LOWER_LIMIT_WARNING_MODEReference

Usage example

```
Dim vUpperLowerLimitWarningMode As Variant
vUpperLowerLimitWarningMode = caoCtrl.Execute("GetUpperLowerLimitWarningMode")
```

3.1.18. CaoController::Execute("GetBuzzerSoundProcess") command

Acquire the buzzer sound processing of the NEO series.

Format GetBuzzerSoundProcess

Argument : None
Return value : Buzzer sound processing (VT_I4)
@BUZZER_SOUND_PROCESSReference

Usage example

```
Dim vBuzzerSoundProcess As Variant
vBuzzerSoundProcess = caoCtrl.Execute("GetBuzzerSoundProcess")
```

3.1.19. CaoController::Execute("GetTemperatureSettingRange") command

Acquire the range of the temperature setting of the NEO series.

Format GetTemperatureSettingRange

Argument : None
Return value : Range of temperature setting (VT_ARRAY | VT_R8)
@TEMPERATURE_SETTING_RANGEReference

Usage example

```
Dim vTemperatureSettingRange As Variant
vTemperatureSettingRange = caoCtrl.Execute("GetTemperatureSettingRange")
```

3.1.20. CaoController::Execute("GetHumiditySettingRange") command

Acquire the range of the humidity setting of the NEO series.

Format GetHumiditySettingRange

Argument : None
Return value : Range of humidity setting (VT_ARRAY | VT_R8)

@HUMIDITY_SETTING_RANGEReference

Usage

example

```
Dim vHumiditySettingRange As Variant
vHumiditySettingRange = caoCtrl.Execute("GetHumiditySettingRange")
```

3.1.21. CaoController::Execute("GetHumidityControlledTemperatureRange") command

Acquire the range of the NEO series of the humidity control temperature.

Format GetHumidityControlledTemperatureRange

Argument : None

Return value : Range of humidity control temperature (VT_ARRAY | VT_R8)
@HUMIDITY_CONTROLLED_TEMPERATURE_RANGEReference

Usage

example

```
Dim vHumidityControlledTemperatureRange As Variant
vHumidityControlledTemperatureRange = _
 caoCtrl.Execute("GetHumidityControlledTemperatureRange")
```

3.1.22. CaoController::Execute("GetUpperLowerLimitTemperatureHumdity") command

Acquire the bound pair temperature humidity of the NEO series.

Format GetUpperLowerLimitTemperatureHumidity

Argument : None

Return value : Bound pair temperature humidity (VT_ARRAY | VT_I4)
@UPPER_LOWER_LIMIT_TEMPERATURE_HUMIDITYReference

Usage

example

```
Dim vUpperLowerLimitTemperatureHumidity As Variant
vUpperLowerLimitTemperatureHumidity = _
 caoCtrl.Execute("GetUpperLowerLimitTemperatureHumidity")
```

3.1.23. CaoController::Execute("GetProgramData") command

Acquire the program data of the NEO series.

Format GetProgramData

Argument	:	Step No.0-49 (VT_I4)
Return value	:	Program data (VT_ARRAY | VT_VARIANT) @PROGRAM_DATAReference

Usage example

```
Dim vProgramData As Variant
Dim vParam As Variant
vParam = 0&
vProgramData = caoCtrl.Execute("GetProgramData", vParam)
```

3.1.24. CaoController::Execute("GetRepeatData") command

Acquire the repetition data of the NEO series.

Format GetRepeatData

Argument	:	None
Return value	:	Repetition data (VT_ARRAY | VT_VARIANT) @REPEAT_DATAReference

Usage example

```
Dim vRepeatData As Variant
```

```
vRepeatData = caoCtrl.Execute("GetRepeatData")
```

3.1.25. CaoController::Execute("GetHumidificationDelay") command

Acquire the humidifying delay of the NEO series.

Format GetHumidificationDelay

Argument	:	None
Return value	:	Humidifying delay (VT_I4) @HUMIDIFICATION_DELAYReference

Usage example

```
Dim vHumidificationDelay As Variant
vHumidificationDelay = caoCtrl.Execute("GetHumidificationDelay")
```

3.1.26. CaoController::Execute("GetTestEndOperation") command

Acquire the examination end driving of the NEO series.

Format GetTestEndOperation

Argument	:	None
Return value	:	Examination end driving (VT_I4) @TEST_END_OPERATIONReference

Usage example

```
Dim vTestEndOperation As Variant
vTestEndOperation = caoCtrl.Execute("GetTestEndOperation")
```

3.1.27. CaoController::Execute("GetAutoStartTime") command

Acquire the auto start time of the NEO series.

Format GetAutoStartTime

Argument : None
Return value : Auto start time (VT_ARRAY | VT_UI2)
@AUTO_START_TIMEReference

Usage
example

```
Dim vAutoStartTime As Variant
vAutoStartTime = caoCtrl.Execute("GetAutoStartTime")
```

3.1.28. CaoController::Execute("GetStartStep") command

Acquire the start step of the NEO series.

Format GetStartStep

Argument : None
Return value : Start step (VT_I4)
@START_STEPReference

Usage
example

```
Dim vStartStep As Variant
vStartStep = caoCtrl.Execute("GetStartStep")
```

3.1.29. CaoController::Execute("GetAutoStopRemainingTime") command

Acquire the auto stop remainder time of the NEO series.

Format GetAutoStopRemainingTime

Argument : None
Return value : The auto stop remainder time (VT_ARRAY | VT_UI2)

@AUTO_STOP_REMAINING_TIMEReference

Usage

example

```
Dim vAutoStopRemainingTime As Variant
vAutoStopRemainingTime = caoCtrl.Execute("GetAutoStopRemainingTime")
```

4. Error code

In the ETAC NEO COM provider, an error code that as follows and is peculiar is defined.

About the ORiN2 commonness error code, please refer to the chapter of the error code of "ORiN2SDK programing guide"..

Table 4—1 peculiar error code

Error name	Error number	Explanation
Receive data is abnormal.	0x80100000	When abnormality is found in receive data, it is returned.
The parameter number is abnormal.	0x80100001	When abnormality is found in the number of parameters, it is returned.
The mnemonic is abnormal.	0x80100002	When it is an illegal mnemonic, it is returned.
The data order is abnormal.	0x80100003	When abnormality is found in the order of data, it is returned.
Character outside regulations	0x80100101	Error reply Code 1 Mnemonic ILLCHAR Character outside content regulations When [wo] is received, it is returned.
Can't be omitted	0x80100102	Error reply Code 2 Mnemonic MUST The content cannot be omitted. When [wo] is received, it is returned.
The parameter type is incompatible.	0x80100103	Error reply Code 3 Mnemonic ILLPARAM The content parameter type is incompatible. When [wo] is received, it is returned.
Parameter number excess	0x80100105	Error reply Code 5 Mnemonic TOOMANY Content parameter number excess

		When [wo] is received, it is returned.
It underestimates the parameter number.	0x80100106	Error reply Code 6 Mnemonic TOOFEW It underestimates the content parameter number. When [wo] is received, it is returned.
Command outside regulations	0x80100107	Error reply Code 7 Mnemonic UNKCMD Command outside content regulations When [wo] is received, it is returned.
Internal error	0x80100108	Error reply Code 8 Mnemonic INTERNAL Internal error of content When [wo] is received, it is returned.
Outside parameter range	0x8010010A	Error reply Code 10 Mnemonic OUTOFRANGE Outside content parameter range When [wo] is received, it is returned.
Port difference	0x8010010B	When the variable or the command is used it is not possible to use it in the port under the connection, it is returned.

5. Sample program

The sample using the ETAC NEO COM provider is shown.

List 5-1 Sample.frm

```
Option Explicit

Private m_Engine As CaoEngine
Private m_Workspace As CaoWorkspace
Private m_Controller As CaoController
Private m_Variable As CaoVariable

Private Sub cmdExecute_Click()

 '  execute GetDeviceInfo.
 Dim vDeviceInfo As Variant
 vDeviceInfo = m_Controller.Execute("GetDeviceInfo")
 txtMaker.Text = vDeviceInfo(0)
 txtModel.Text = vDeviceInfo(1)
 txtNo.Text = vDeviceInfo(2)
 txtVersion.Text = vDeviceInfo(3)

End Sub

Private Sub cmdValue_Click()

Acquire the value of a variable of '@DEVICE_INFO.
 txtMaker.Text = m_Variable.Value(0)
 txtModel.Text = m_Variable.Value(1)
 txtNo.Text = m_Variable.Value(2)
 txtVersion.Text = m_Variable.Value(3)

End Sub

Private Sub Form_Load()

'  connection
 Set m_Engine = New CaoEngine
 Set m_Workspace = m_Engine.Workspaces(0)
 Set m_Controller = m_Workspace.AddController("EtacNeoCom", "CaoProv.ETAC.NEO_COM", _
 "", "Conn=ETH:192.168.0.1:10052,ConnTimeout=500,Timeout=3000")
 CaoVariable '  addition
 Set m_Variable = m_Controller.AddVariable("@DEVICE_INFO")

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

'  cutting
 m_Controller.Variables.Remove m_Variable.Index
 Set m_Variable = Nothing
 m_Workspace.Controllers.Remove m_Controller.Index
 Set m_Controller = Nothing
 Set m_Workspace = Nothing
 Set m_Engine = Nothing

End Sub
```