CVX Provider
KEYENCE CV-X series

Version 1.0.3

User's guide

August 29, 2019

[Remarks]

[Revision history]

Version	Data	Content
1.0.0	2014-02-11	First edition.
1.0.1	2016-01-20	Added Asynchronous processing
1.0.2	2017-07-05	Added CV-X200 controller-supported commands ・ResetTrigger ・ReadMode ・SetShutterSpeed ・SetSensitivity ・SetTriggerDelay ・EnterImg ・UpdateCapturePos
1.0.3	2019-07-09	Added Measurement commands. ・RegisterOneCharToLibrary ・DeleteOneCharFromLibrary Added Measured value correction commands. ・ChangePreCorrectionMeasuredValue ・WriteMeasuredValueCorrection ・ReadMeasuredValueCorrection
1.0.3	2019-08-29	Added precautions for the use of asynchronous command.

[Supported models]

Model	Version	Note
CVX-100		
CVX-200		
CVX-400		
CVX-480		

Contents

1. Introduction

This document is a user's guide of CVX provider that is CAO provider designed for the vision system "CVX series" manufactured by KEYENCE.

CVX provider executes non-procedure commands and also notifies transaction results to CVX series controller through Ethernet or RS-232C

This document describes functions of CVX provider and equipped methods.

1.1. Device setup

In CVX provider, RS-232C and Ethernet are available for communication. Select a desired communication setting from the Global in the menu bar.

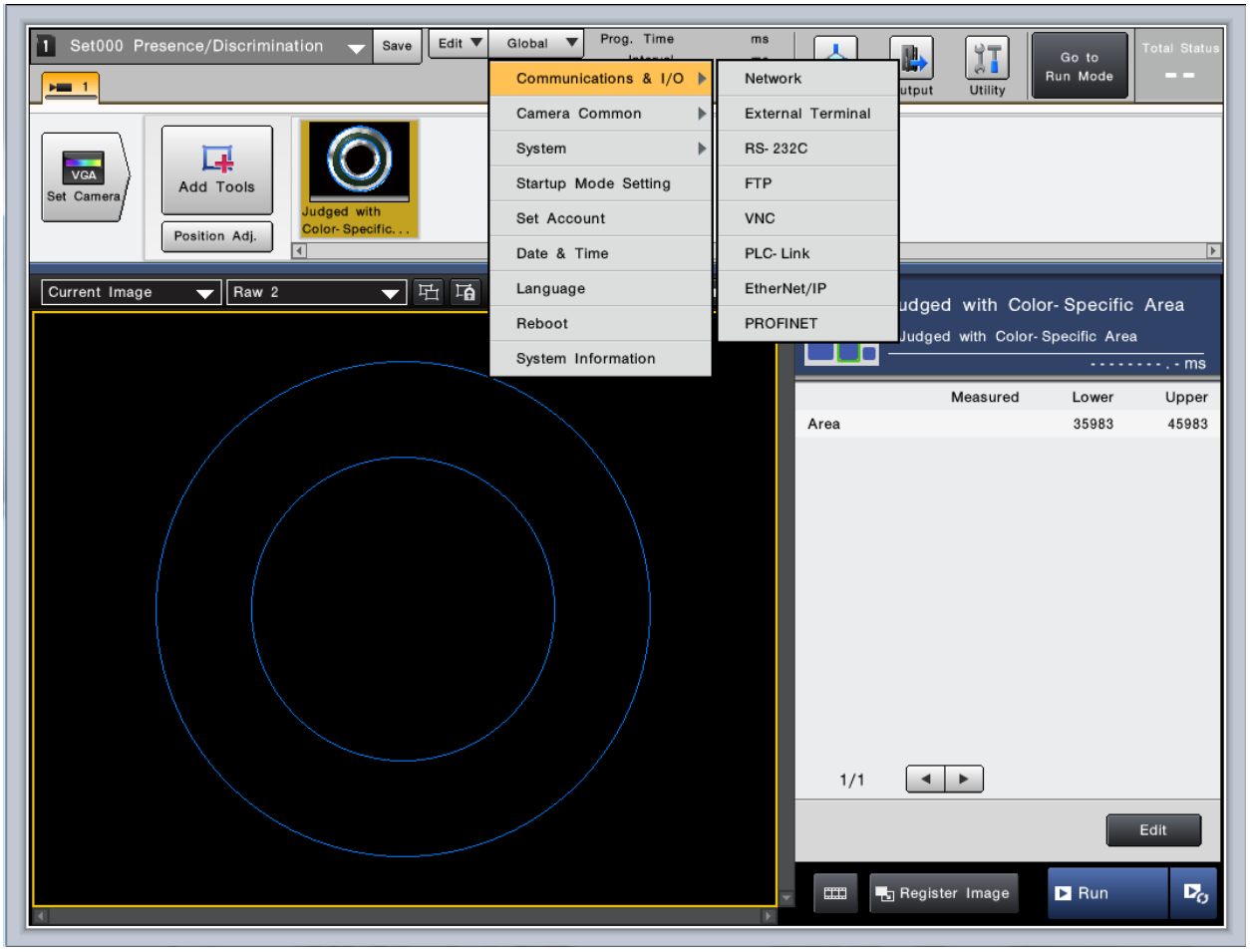

Figure 1-1 Global setting

1.1.1. Setup for RS-232C connection

In RS-232C communication setting, all item settings other than the Delimiter can be changed. If you change any settings from the default values, you need to change the connection parameters of AddController as well (see 2.2.1.1). Select CR for the Delimiter.

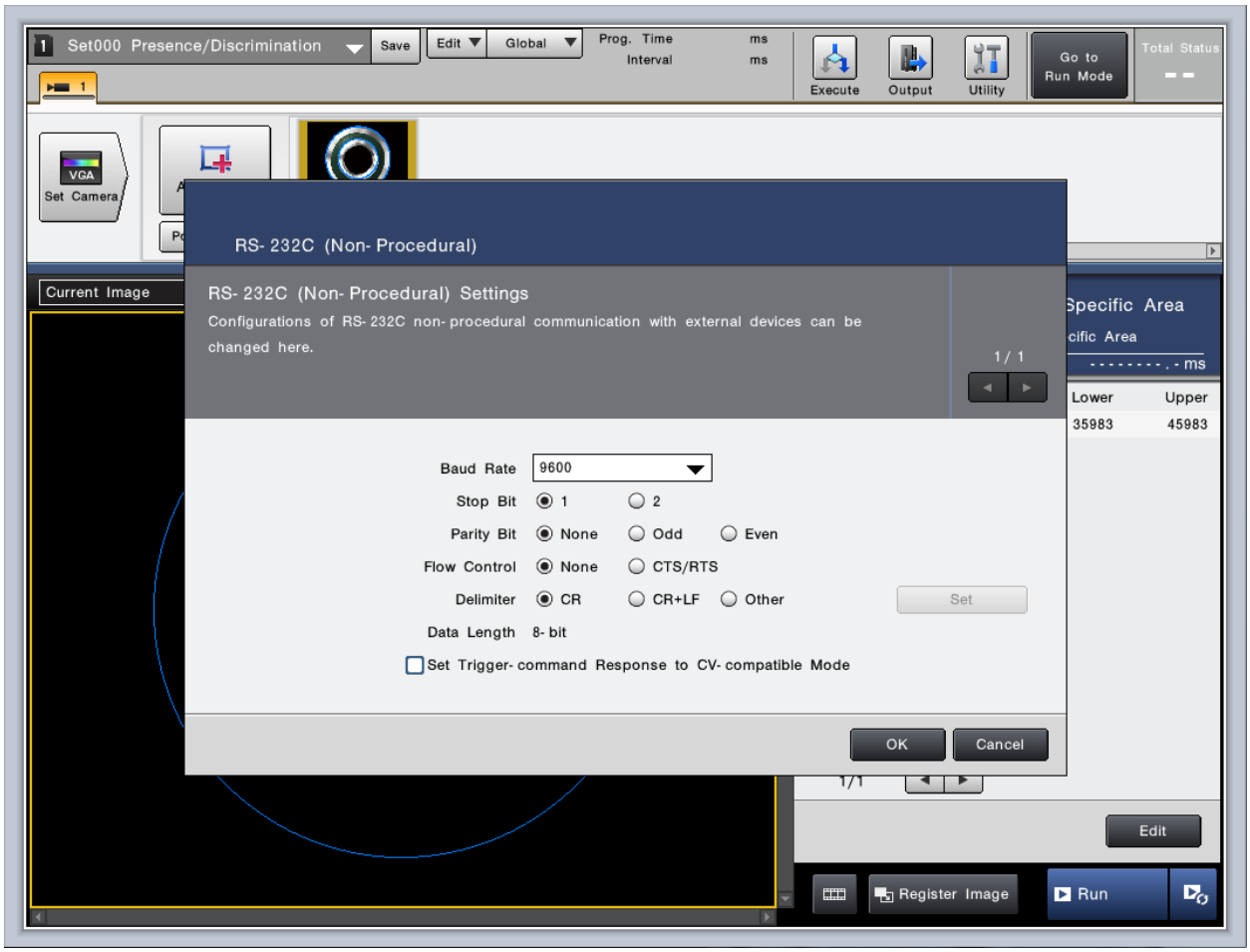

Figure 1-2 RS-232C setting

1.1.2. Setup for Ethernet connection

In Ethernet connection setting, all items other than the Delimiter can be changed. If you change any settings from the default values, you need to change the connection parameters of AddController as well (see 2.2.1.1). Select CR for the Delimiter.

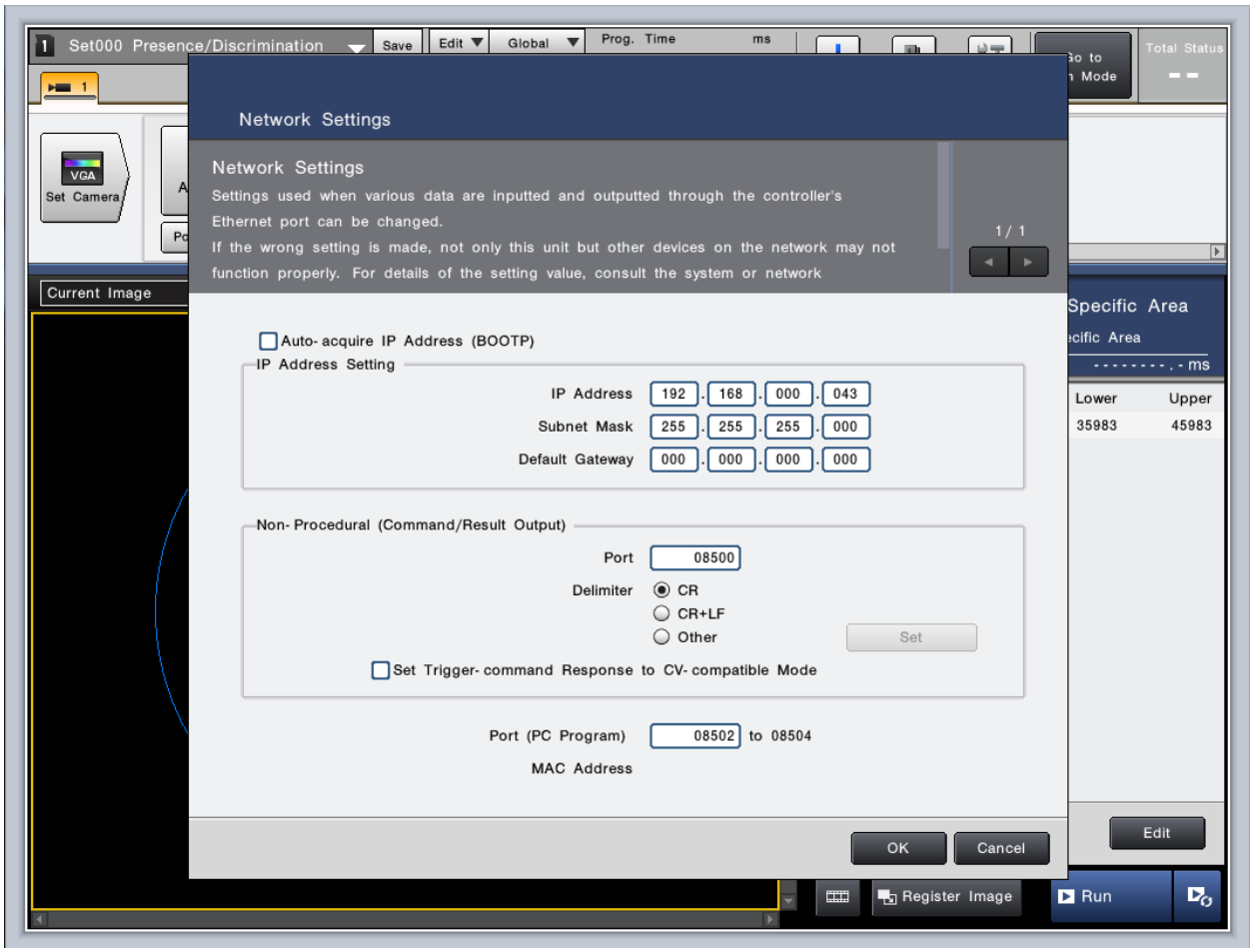

Figure 1-3 Network settings

2. Overview of provider

2.1. Overview

CVX provider provides CaoController::Execute methods for a command execution.

In CaoController::Execute, commands are sent or received through a serial interface.

Table 2-1 CVX provider

File name	CaoProvCVX.dll
ProgID	CaoProv.KEYENCE.CVX
Registry registration[1]	regsvr32 CaoProvCVX.dll
Remove registry registration	regsvr32 /u CaoProvCVX.dll

2.2. Method and Property

2.2.1. CaoWorkspace::AddController method

CVX provider establishes communication by referring to the connection parameters for communication when AddController is executed. At that time, communication method is specified in option.

Syntax AddController(<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])

bstrCtrlName : [in] Controller name (any name)
bstrProvName : [in] Provider name (Fixed to “CaoProv.KEYENCE.CVX”)
bstrPcName : [in] Provider execution machine name
bstrOption : [in] Option character string

Following is a list of option string items.

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

Option	Explanation
Conn =<Connection parameter>	This must be specified. Set communication configuration and its connection parameter. For details, see 2.2.1.1 .
Timeout[=<Timeout period>]	Specify a timeout period (millisecond) at the receiving and sending. (Default: 500)

[1] If the registry is installed by ORiN SDK, you do not need to register/delete it manually.

2.2.1.1. Conn option

Following shows connection parameter strings for Conn option. Parameters surrounded by the square brackets (“[]”) can be omitted. Underlined part shows the default value when the option is not specified

- Ethernet device

  "eth:<IP Address> [:<Port No>]"

  <IP Address> : IP address of CVX series being connected

  Example: ”192.168.0.10”, “192.168.0.11”

  <Port No> : Connection port number

  8500, 8501, … Arbitrary number can be specified

Example

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng  = New CaoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("CVX", "caoProv.KEYENCE.CVX", "",
"conn=eth:192.168.0.1, timeout=800")
```

- RS-232C device

  "Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"

  <COM Port> : COM port number

  ‘1’-COM1, ‘2’-COM2, …

  <BaudRate> : Baud rate

  4800, 9600, 19200, 38400, 57600, 115200.

  <Parity> : Parity

  ‘N’-NONE, ‘E’-EVEN, ‘O’-ODD.

  <DataBits> : Number of the data bit

  ‘7’-7bit, ‘8’-8bit.

  <StopBits> : Number of the stop bit

  ‘1’-1bit, ‘2’-2bit.

  <Flow> : Flow control

  ‘0’-Without flow control, ‘1’-Xon/Xoff, ‘2’-Hardware control

Example

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng  = New CaoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("CVX", "caoProv.KEYENCE.CVX", "",
"conn=com:1")
```

2.2.2. CaoController::Execute method

CVX provider sends and receives command with non-procedure methods through Ethernet. Set a command name to the first argument, a command parameter to the second argument. For details about each command, see section 3. Command reference.

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

bstrCommandName: [in] Command name

vntParam : [in] Parameter

2.2.3. Error code

The processing result from the CVX series at the method execution is returned as HRESULT. CVX series-specific errors are masked with 0x80108000 and returned. For details about CVX series-specific errors, see CVX series user's guide.

When the process completed successfully (OK): S_OK (0)

When the process completed with errors (ER): 0x80108000 + Return value

Example:When ChangeMode is executed.

hr = 0x80108016: Either the number or content of the parameters is incorrect

For details about other provider-common errors, refer to the ORiN2SDK Programmers User's Guide.

Table 2-3 Error code list

Error	Error number	Description
E_CVERROR_CVERR	0x80108000 | CVX error	CVX series original error (see 2.2.3)
E_CVERROR_LENGTH	0x80100000	Packet length error
E_CVERROR_PACKET	0x80100001	Packet error
E_COMMAND_EXECUTING	0x80100002	Another command was executed during a command execution
E_GET_COMMAND_RESULT	0x80100003	GetCommandResult command was executed after a Synchronous command

		execution

3. Command reference

This chapter describes each command of CaoController::Execute method. For information about motion detail, parameter range, description of parameter of each command, please refer to the communication control command reference of KEYENCE CVX-series user's manual.

Table 3-1 CaoController::Execute command list

CVX series Non-procedure command	Command	Function	
Trigger			
T1, T2, T3, T4, TA	Trigger	Inputs a trigger	P.13
System control			
R0, S0	ChangeMode	Changes the operation mode	P.14
	ChangeModeAsync	Changes the operation mode asynchronously. (not recommended)	P.14
RS	Reset	Reset.	P.15
RB	Reboot	Saves the current program and reboot the system	P.15
SS	StoreSetting	Saves the program setting	P.16
CE	ClearError	Clears errors	P.16
VW	ChangeDisplayPattern	Changes the display pattern	P.16
TE	EnableTrigger	Enables/Disables trigger input	P.17
RE	ResetTrigger	Discards the measurement and goes back to the state before measurement.	P.17
RM	ReadMode	Reads the controller status.	P.17
Change program settings			
PW	ChangeInspectSetting	Switches the program number	P.18
	ChangeInspectSettingAsync	Switches the program number asynchronously (not recommended)	P.18
PR	ReadInspectSetting	Reads the program number	P.19
Capture control			
CLV	ChangeLightVolume	Changes the intensity of the specified lighting	P.19
CSH	SetShutterSpeed	Changes the shutter speed.	P.20
CSE	SetSensitivity	Changes the sensitivity.	P.20
CTD	SetTriggerDelay	Places a delay time between the trigger input and the start of image capture.	P.21

Vision tool			
EXW	ChangeExecuteCondition	Writes the execute condition	P.21
EXR	ReadExecuteCondition	Reads the execute condition	P.21
CW	WriteCharReg	Changes the registered string	P.22
CR	ReadCharReg	Reads the registered string	P.22
DW	ChangeToolParameter	Changes the judgment condition	P.23
DR	ReadToolParameter	Reads the judgment condition	P.23
SLW	ChangeFlawLevel	Changes the stain level	P.23
SLR	ReadFlawLevel	Reads the stain level	P.24
BS	EnterImg	Changes the reference image and calculates the reference value.	P.24
CPW	UpdateCapturePos	Updates the capture position coordinates.	P.25
CA	RegisterOneCharToLibrary	Adds one character to the library	P.25
CD	DeleteOneCharFromLibrary	Deletes one character from the library.	P.26
Measured value correction			
MCC	ChangePreCorrectionMeasuredValue	Calculates the measured value before correction for the specified value.	P.26
MCW	WriteMeasuredValueCorrection	Changes the measured value correction setting of the specified measurement item number for a given tool and then calculates the correction value.	P.27
MCR	ReadMeasuredValueCorrection	Returns the value for Measured value correction set in the specified tool.	P.28
Original extended command			
-	ExecuteCommand	Executes a Non-procedure command	P.29
-	ExecuteCommandAsync	Executes a Non-procedure command asynchronously (not recommended)	P.30
-	TriggerAndGetResult	Issues a trigger and receive the output result	P.30
-	RecievePacket	Receives packets.	P.31
-	ClearPacket	Clears the receiving packet in the buffer	P.31
-	SetTimeout	Sets the timeout period	P.31
-	GetTimeout	Gets the timeout period	P.32
-	GetCommandResult	Gets the return value of asynchronous command	P.32

3.1. Trigger

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

Issues a trigger. To receive processed data, use RecievePacket command or others.

Syntax Trigger(< iTriggerNo >)

iTriggerNo : Specify a trigger number (VT_I2)
1 to 4 : Trigger number
-1: All triggers

Return value : none

Executing the following example will issue a trigger number 1.

Example

```
caoCtrl.Execute "Trigger", 1
```

3.2. System control

3.2.1. CaoController::Execute ("ChangeMode") command

Changes to Run mode or Program mode.

Syntax ChangeMode(< iMode >)

iMode : [in] Specify a desired mode (VT_UI4)
0 : Program mode
1 : Run mode

Return value : none

Executing the following example will switch the mode to Run mode.

Example

```
caoCtrl.Execute "ChangeMode", 1
```

3.2.2. CaoController::Execute ("ChangeModeAsync") command (not recommended)

Changes the operation mode (Run or Program mode) asynchronously

To obtain and check the return value of the command, use GetCommandResult command.For details about GetCommandResult command, refer to 3.7.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ChangeModeAsync(< iMode >)

iMode : [in] Specify a desired mode (VT_UI4)
0 : Program mode
1 : Run mode

Return value : None

Executing the following example will switch the mode to Run mode.

Example

```
Dim vntResult as Variant

caoCtrl.Execute "ChangeModeAsync", 1

' Obtain the return value of ChangeModeAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value(Empty)
```

3.2.3. CaoController::Execute ("Reset") command

Resets items, such as image buffer, history data, etc. For details, see CVX series user's manual.

Syntax Reset

Argument : none

Return value : none

Example

```
caoCtrl.Execute "Reset"
```

3.2.4. CaoController::Execute ("Reboot") command

Saves the current program settings, and reboot the system.

Syntax Reboot

Argument : none

Return value : none

Example

```
caoCtrl.Execute "Reboot"
```

3.2.5. CaoController::Execute ("StoreSetting") command

Saves currently selected program settings and global settings

Syntax StoreSetting

Argument : none
Return value : none

Example

```
caoCtrl.Execute "StoreSetting"
```

3.2.6. CaoController::Execute ("ClearError") command

Clears the error status. Even when an error status does not exist, the command execution finishes normally.

Syntax ClearError

Argument : none
Return value : none

Example

```
caoCtrl.Execute "ClearError"
```

3.2.7. CaoController::Execute ("ChangeDisplayPattern") command

Changes the display to the specified operation screen.

Syntax ChangeDisplayPattern(<lDisplay>, <lNum>)

lDisplay : Specified screen type (VT_I4)
lNum : Screen number (VT_I4)
Return value : none

Executing the following example will change the image display to the camera number 1

Example

```
caoCtrl.Execute "ChangeDisplayPattern", Array(0, 1)
```

3.2.8. CaoController::Execute ("EnableTrigger") command

Enables/Disables trigger input

Syntax EnableTrigger(< iMode >)

iMode	:	[in] Enable/Disable a trigger. (VT_I2)
Return value	:	none

Executing the following example will disable the trigger input

Example

```
caoCtrl.Execute "EnableTrigger", 0
```

3.2.9. CaoController::Execute ("ResetTrigger") command

Discards the currently captured image and measurement results and goes back to the state before the measurement.

Syntax ResetTrigger

Argument	:	[in] none
Return value	:	[out] none

The following shows the sample program.

Example

```
caoCtrl.Execute "ResetTrigger"
```

3.2.10. CaoController::Execute ("ReadMode") command

Reads-out the controller status (run mode/setup mode).

Syntax ReadMode

Argument	:	[in] none
Return value	:	[out] controller state (VT_I4)

The following shows the sample program.

Example

```
caoCtrl.Execute "ReadMode"
```

3.3. Change program settings

3.3.1. CaoController::Execute ("ChangeInspectSetting") command

Changes the setting to the inspection setting number of the specified SD card.

Syntax ChangeInspectSetting (< iDriveNo >, < iSettingNo >)

iDriveNo	:	[in] Specify an SD card number(VT_I4)
iSettingNo	:	[in] Specify an inspection setting number (VT_I4)
Return value	:	none

Executing the following example will change the setting to the inspection setting number 1 of the SD card (SD1)

Example

```
call caoCtrl.Execute("ChangeInspectSetting", Array(1, 1))
```

3.3.2. CaoController::Execute ("ChangeInspectSettingAsync") command (not recommended)

Asynchronously changes the setting to the inspection setting number of the specified SD card.

To obtain and check the return value of the command, use GetCommandResult command.For details about GetCommandResult command, refer to 3.7.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ChangeInspectSettingAsync (< iDriveNo >, < iSettingNo >)

iDriveNo	:	[in] Specify SD card number (VT_I4)
iSettingNo	:	[in] Specify an inspection setting number (VT_I4)
Return value	:	none

Executing the following example will change the setting to the inspection setting number 1 of the SD card (SD1)

Example

```
Dim vntResult as Variant

call caoCtrl.Execute("ChangeInspectSettingAsync", Array(1, 1))

' Obtain the return value of ChangeInspectionSettingAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

vntResult : Return value(Empty)
```

3.3.3. CaoController::Execute ("ReadInspectSetting") command

Reads the inspection setting number

Syntax ReadInspectSetting ()

Argument	:	[in] none
Return value	:	[out] < iDriveNo >, < iSettingNo > (VT_I4 | VT_ARRAY)
		iDriveNo : SD card number
		iSettingNo : Inspection setting number

Executing the following example will read the current inspection number.

Example

```
Dim vntRet as Variant
vntRet = caoCtrl.Execute("ReadInspectSetting")
' vntRet(0) : SD card number
' vntRet(1) : Inspection setting number
```

3.4. Capture control

3.4.1. CaoController::Execute ("ChangeLightVolume") command

Changes the intensity of the specified lighting

Syntax ChangeLightVolume(<lLightNo>, <lVolume> [, <lCnt> [, <lNum>]])

lLightNo	:	[in] Lighting number (VT_I4)
lVolume	:	[in] Lighting intensity value (VT_I4)
lCnt	:	[in] Capture count or capture point (VT_I4)
lNum	:	[in] Lighting for multiple image capture (VT_I4)
Return value	:	[out] none

Executing the following example will set the lighting intensity value of lighting number 1 to 30.

Example

```
caoCtrl.Execute "ChangeLightVolume", Array(1, 30)
```

3.4.2. CaoController::Execute ("SetShutterSpeed") command

Changes the shutter speed for the specified camera.

Syntax SetShutterSpeed(<lCameraNo>, <lShutterSpeed> [, <lCnt> [, <lNum>]])

lCameraNo	:	[in] Camera No. (VT_I4)
lShutterSpeed	:	[in] Shutter speed (VT_I4)
lCnt	:	[in] Capture number or capture point (VT_I4)
lNum	:	[in] Lighting for multiple image capture (VT_I4)
Return value	:	[out] none

Executing the following sample will set the shutter speed of Camera1 to “5”.

Example

```
caoCtrl.Execute "SetShutterSpeed",Array(1,5)
```

3.4.3. CaoController::Execute ("SetSensitivity") command

Changes the sensitivity of the specified camera

Syntax SetSensitivity(<lCameraNo>, <lSensitivity> [, <lCnt> [, <lNum>]])

lCameraNo	:	[in] Camera No. (VT_I4)
lSensitivity	:	[in] Sensitivity (VT_I4)
lCnt	:	[in] Capture number or capture point (VT_I4)
lNum	:	[in] Lighting for multiple image capture (VT_I4)
Return value	:	[out] none

Executing the following sample will set the sensitivity of Camera1 to “50”.

Example

```
caoCtrl.Execute "SetSensitivity",Array(1,50)
```

3.4.4. CaoController::Execute ("SetTriggerDelay") command

Places a delay time between the trigger input and the start of image capture.

Syntax SetTriggerDelay(<lCameraNo>, <lTriggerDelay> [, <lCnt> [, <lNum>]])

lCameraNo	:	[in] Camera No. (VT_I4)
lTriggerDelay	:	[in] Trigger delay (msec) (VT_I4)
lCnt	:	[in] Capture number or capture point (VT_I4)
lNum	:	[in] Lighting for multiple image capture (VT_I4)
Return value	:	[out] none

Executing the following sample will place 500 milliseconds delay between the trigger input and the start of image capture of Camera 1.

Example

```
caoCtrl.Execute "SetTriggerDelay",Array(1,500)
```

3.5. Vision tool

3.5.1. CaoController::Execute ("ChangeExecuteCondition") command

Changes the number of the execution condition currently enabled to the specified number.

Syntax ChangeExecuteCondition (< lNo >)

lNo	:	[in] Execute condition number (VT_I4)
Return value	:	[out] none

Executing the following program will set the execute condition number to 1.

Example

```
caoCtrl.Execute "ChangeExecuteCondition", 1
```

3.5.2. CaoController::Execute ("ReadExecuteCondition") command

Reads the number of the executable condition currently enabled.

Syntax ReadExecuteCondition()

Argument	:	[in] none
Return value	:	[out] Execute condition number (VT_I4)

Example

```
Dim lParam as Long
lParam = caoCtrl.Execute("ReadExecuteCondition")
```

3.5.3. CaoController::Execute ("WriteCharReg") command

The No. nnn OCR tool registered string and the reference pattern string for the 1D code reader tool and 2D code reader tool are changed to the specified character string ssss. If registered string ssss is not specified, the most recent reading result for that tool is set.

Syntax WriteCharReg (< lToolNo >, <lRowNo>[, <bstrParam>])

lToolNo	:	[in] Tool number (VT_I4)
lRowNo	:	[in] Line No./Reference condition No.(VT_I4)
bstrParam	:	[in] Registered string (VT_BSTR)
Return value	:	[out] none

Executing the following program will set the No.101 OCR tool registered string to DEF.

Example

```
caoCtrl.Execute "WriteCharReg", Array(101,1,"DEF")
```

3.5.4. CaoController::Execute ("ReadCharReg") command

Reads the No. nnn OCR tool registered string, and the reference pattern string of the 1D code reader tool and 2D code reader tool,

Syntax ReadCharReg (< lToolNo >, <lRowNo>[, <bstrParam>])

lToolNo	:	[in] Tool number (VT_I4)
lRowNo	:	[in] Line No./Reference condition No (VT_I4)
Return value	:	[out] Registered string (VT_BSTR)

Executing the following program will obtain the No.101 OCR tool registered string.

Example

```
Dim strParam as String
strParam = caoCtrl.Execute("ReadCharReg", Array(101,1))
```

3.5.5. CaoController::Execute ("ChangeToolParameter") command

Changes the upper limit and lower limit in the judgment conditions for the specified tool.

Syntax ChangeToolParameter (< lToolNo >, <lID>, <lFlag>, <vntParam>)

lToolNo	:	[in] Tool number (VT_I4)
lID	:	[in] Item ID for judgment condition type (VT_I4)
lFlag		[in] Specify upper limit (0) or lower limit (1) (VT_I4)
vntParam		[in] Judgment condition value (VT_VARIANT)
Return value	:	[out] none

Executing the following program will set the lower limit on the edge tool of the tool No.100 to 123.456.

Example

```
caoCtrl.Execute "ChangeToolParameter", Array(100,82,1,"123.456")
```

3.5.6. CaoController::Execute ("ReadToolParameter") command

Reads the upper limit and lower limit in the judgment conditions for the specified tool.

Syntax ReadToolParameter (< lToolNo >, <lID>, <lFlag>)

lToolNo	:	[in] Tool number (VT_I4)
lID	:	[in] Item ID for judgment condition type (VT_I4)
lFlag		[in] Specify upper limit (0) or lower limit (1) (VT_I4)
Return value	:	[out] Judgment condition value (VT_BSTR)

Executing the following program will read the lower limit on the edge tool of the tool No.100.

Example

```
Dim strParam as String
strParam = caoCtrl.Execute("ReadToolParameter", Array(100,82,1))
```

3.5.7. CaoController::Execute ("ChangeFlawLevel") command

Changes the stain level for the specified stain tool

Syntax ChangeFlawLevel (< lToolNo >, <vntParam>)

lToolNo	:	[in] Tool number (VT_I4)
vntParam	:	[in] Stain level value (VT_I4)
Return value	:	[out] none

Executing the following program will change the stain level value of the tool number 102 to 200.

Example

```
caoCtrl.Execute "ChangeFlawLevel", Array(102,200)
```

3.5.8. CaoController::Execute ("ReadFlawLevel") command

Reads the stain level of the specified stain tool.

Syntax ReadFlawLevel (< lToolNo >)

lToolNo	:	[in] Tool number (VT_I4)
Return value	:	[out] Stain level value (VT_I4)

Executing the following program will obtain the stain level value of the tool number 102.

Example

```
Dim lParam as Long
lParam = caoCtrl.Execute("ReadFlawLevel", 102)
```

3.5.9. CaoController::Execute ("EnterImg") command

Saves the latest current image as a reference image with number lImageNo, and then calculates the reference value based on the saved reference image.

Syntax EnterImg([<lCameraNo>, <lImageNo> [, <lCnt> [, <lNum>]]])

lCameraNo	:	[in] Camera No. (VT_I4)
lImageNo	:	[in] Reference image No. (VT_I4)
lCnt	:	[in] Capture number or capture point (VT_I4)
lNum	:	[in] Lighting for multiple image capture (VT_I4)
Return value	:	[out] none

Executing the following program will save the latest input image of Camera 1 as the reference image of

number 500.

Example

```
caoCtrl.Execute "EnterImg",Array(1,500)
```

3.5.10. CaoController::Execute ("UpdateCapturePos") command

Updates the capture position coordinates of the robot vision tool of the specified camera number or of all cameras.

Syntax UpdateCapturePos(<lCameraNo>, <vntPosX> , <vntPosY> , <vntHeight> , <vntAngleX> , <vntAngleY> , <vntAngleZ>)

lCameraNo	:	[in] Camera No. (VT_I4)
vntPosX	:	[in] X position (VT_VARIANT)
vntPosY	:	[in] Y position (VT_VARIANT)
vntHeight	:	[in] Height (VT_VARIANT)
vntAngleX	:	[in] Angle X (VT_ VARIANT)
vntAngleY	:	[in] Angle Y (VT_ VARIANT)
vntAngleZ	:	[in] Angle Z (VT_ VARIANT)
Return value	:	[out] none

Executing the following program will update the latest capture position coordinate of Camera 1.

Example

```
caoCtrl.Execute "UpdateCapturePos",Array(1,500,600,700,0,60,120)
```

3.5.11. CaoController::Execute ("RegisterOneCharToLibrary") command

Registers characters read by the OCR tool or OCR2 tool to the OCR library.

Syntax RegisterOneCharToLibrary (<lToolNo>, <lRowNo>, <lCharNo>, <lCharType>)

lToolNo	:	[in] Tool No. (VT_I4)
lRowNo	:	[in] Detect result row number (VT_I4)
lCharNo	:	[in] Detect result character number (VT_I4)
lCharType	:	[in] Destination character type (VT_I4)

Return value : [out] none

Executing the following program will read the first character on the first row by OCR tool (tool number 100) and then register it in the library with character type 2.

Example

```
caoCtrl.Execute "RegisterOneCharToLibrary",Array(100,1,1,2)
```

3.5.12. CaoController::Execute ("DeleteOneCharFromLibrary") command

Deletes the character for the last Register No. of the specified character type from the library.

Syntax DeleteOneCharFromLibrary (<lToolNo>, <lCharType>)

lToolNo : [in] Tool No. (VT_I4)

lCharType : [in] Type of character to be deleted (VT_I4)

Return value : [out] none

Executing the following program will delete a character that has been registered with character type 2 at the latest in the library by OCR tool (tool number 100).

Example

```
caoCtrl.Execute "DeleteOneCharFromLibrary",Array(100,2)
```

3.6. Measured value correction

3.6.1. CaoController::Execute ("ChangePreCorrectionMeasuredValue") command

Calculates the measured value before correction for the specified value.

Syntax ChangePreCorrectionMeasuredValue (<lToolNo>, <lNo>, <fParam>)

lToolNo : [in] Tool No. (VT_I4)

lNo : [in] Measurement item number (VT_I4)

fParam : [in] Value (VT_R4)

Return value : [out] Correction value (VT_R4)

Executing the following program will calculate the value before correction of specified value "5" by using the conditions of measurement item 1 with Height measurement tool (tool number 101).

Example

```
Dim fParam as Single
fParam = caoCtrl.Execute "ChangePreCorrectionMeasuredValue",Array(101,1,1,2)
```

3.6.2. CaoController::Execute ("WriteMeasuredValueCorrection") command

Changes the measured value correction setting of the specified measurement item number for a given tool and also calculates the correction value.

There are four types of calculation methods. According to the following instruction, specify necessary parameters for each calculation method.

① With 1-point correction method, specify the values before and after the correction to calculate the correction value.

Syntax WriteMeasuredValueCorrection (<lToolNo>, <bstrNo>, <lType>, <fBeforeValue>, <fAfterValue>)

lToolNo	:	[in] Tool No. (VT_I4)
bstrNo	:	[in] Measurement item number (VT_BSTR) ※To specify all measurement items, enter "AL".
lType	:	[in] Correction method (VT_I4) ※0 : 1-point correction
fBeforeValue	:	[in] Value before correction (VT_R4)
fAfterValue	:	[in] Value after correction (VT_R4)
Return value	:	[out] none

② With 1-point correction method, subtract the specified offset value from the set value after correction, and then set this result as the correction value.

Syntax WriteMeasuredValueCorrection (<lToolNo>, <bstrNo>, <lType>, <fOffset>)

lToolNo	:	[in] Tool No. (VT_I4)
bstrNo	:	[in] Measurement item number (VT_BSTR) ※To specify all measurement items, enter "AL".
lType	:	[in] Correction method (VT_I4) ※0 : 1-point correction
fOffset	:	[in] Offset value (VT_R4)
Return value	:	[out] none

③ With 2-point correction method, for both Correction 1 and Correction 2, specify the values before and after correction to calculate the correction values.

Syntax WriteMeasuredValueCorrection (<lToolNo>, <bstrNo>, <lType>, <fBeforeValue1>, <fAfterValue1>, <fBeforeValue2>, <fAfterValue2>)

lToolNo	:	[in] Tool No. (VT_I4)
bstrNo	:	[in] Measurement item number (VT_BSTR) ※To specify all measurement items, enter “AL”.
lType	:	[in] Correction method (VT_I4) ※0 : 1-point correction
fBeforeValue1	:	[in] Value before correction1 (VT_R4)
fAfterValue1	:	[in] Value after correction1 (VT_R4)
fBeforeValue2	:	[in] Value before correction2 (VT_R4)
fAfterValue2	:	[in] Value after correction2 (VT_R4)
Return value	:	[out] none

④ With 2-point correction method, perform the inverse calculation with the specified coefficient A and coefficient B and then set this calculation result as the correction value.

Syntax WriteMeasuredValueCorrection (<lToolNo>, <bstrNo>, <lType>, <fCoefficientA>, <fCoefficientB>)

lToolNo	:	[in] Tool No. (VT_I4)
bstrNo	:	[in] Measurement item number (VT_BSTR) ※To specify all measurement items, enter “AL”.
lType	:	[in] Correction method (VT_I4) ※0 : 1-point correction
fCoefficientA	:	[in] Coefficient A (VT_R4)
fCoefficientB	:	[in] Coefficient B (VT_R4)
Return value	:	[out] none

Executing the following program will perform the above-mentioned calculation method ① on the measurement item1 with Height measurement tool (tool number 101).

Example

```
' 1-point correction  Before correction "2.0" => After correction"6.0"
caoCtrl.Execute "WriteMeasuredValueCorrection",Array(101,"1",0,2.0,6.0)
```

3.6.3. CaoController::Execute (“ReadMeasuredValueCorrection”) command

Returns the values for Measured Value Correction set in the specified tool.

Syntax ReadMeasuredValueCorrection (<lToolNo>, <lNo>, <fParam>)

lToolNo : [in] Tool No. (VT_I4)
lNo : [in] Measurement item number (VT_I4)
Return value : [out] (VT_I4 | VT_ARRAY)
① For 1-point correction
0, <fBeforeValue>, <fAfterValue>, <fOffset>
fBeforeValue : Value before correction (VT_R4)
fAfterValue : Value after correction (VT_R4)
fOffset : Value before correction (VT_R4)
② For 2-point correction
1, <fBeforeValue1>, <fAfterValue1>, <fBeforeValue2>, <fAfterValue2>, <fCoefficientA>, <fCoefficientB>
fBeforeValue1 : Value before correction1 (VT_R4)
fAfterValue1 : Value after correction1 (VT_R4)
fBeforeValue2 : Value before correction2 (VT_R4)
fAfterValue2 : Value after correction2 (VT_R4)
fCoefficientA : Coefficient A (VT_R4)
fCoefficientB : Coefficient B (VT_R4)

Executing the following program will set the correction value from the measurement item1 with Height measurement tool (tool number 101).

Example

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute "ReadMeasuredValueCorrection",Array(101,1)
```

3.7. Original extended command

3.7.1. CaoController::Execute ("ExecuteCommand") command

Executes specified non-procedure command. Command response will be obtained regardless of the command execution result.

About Non-procedure commands to be supported, refer to the CVX series user's manual.

Syntax ExecuteCommand(< bstrCommand >)

bstrCommand : [in] Specify character strings of command (VT_BSTR)
Return value : [out] Return the command response (VT_BSTR)

Executing the following example will specify the Non-procedure command and switch CVX to Run mode.

Example

```
Dim strRet as String
strRet = caoCtrl.Execute("ExecuteCommand", "R0")
```

3.7.2. CaoController::Execute ("ExecuteCommandAsync") command (not recommended)

Executes specified non-procedure command asynchronously.

About Non-procedure commands to be supported, refer to the CVX series user's manual.

To obtain and check the return value of the command, use GetCommandResult command. For details about GetCommandResult command, refer to 3.7.8.CaoController::Execute ("GetCommandResult") command.

When you use this command, be sure to disconnect the provider after the execution of GetCommandResult command.

Otherwise, an error may occur at asynchronous operation.

Syntax ExecuteCommandAsync(< bstrCommand >)

bstrCommand	:	[in] Specify character strings of command (VT_BSTR)
Return value	:	[out] none

Executing the following example will specify the Non-procedure command and switch CVX to Run mode.

Example

```
Dim vntResult as Variant

Call caoCtrl.Execute("ExecuteCommandAsync", "R0")

' Obtain the return value of ExecuteCommandAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Return value of non-procedure command(BSTR)
```

3.7.3. CaoController::Execute ("TriggerAndGetResult") command

Receives a result after trigger execution. If no output result will be received, wait until the timeout period passes.

Syntax TriggerAndGetResult(< iTriggerNo >)

iTriggerNo	:	[in] Trigger number 1 to 4 (VT_I2)
Return value	:	[out] Output result (VT_BSTR)

Example

```
Dim strRet as String
strRet = caoCtrl.Execute("TriggerAndGetResult", 1)
```

3.7.4. CaoController::Execute ("RecievePacket") command

Receives packets. If any packets have already been stored in the receiving buffer, packets in the receiving buffer will be obtained.

Syntax RecievePacket ()

Argument : [in] none

Return value : [out] Receiving packet (VT_BSTR)

The following is an example when this command is used together with a trigger command.

Example

```
Dim strRet as String

caoCtrl.Execute "Trigger", 1
strRet = caoCtrl.Execute("RecievePacket")
```

3.7.5. CaoController::Execute ("ClearPacket") command

Deletes packets stored in the receiving buffer.

Syntax ClearPacket( )

Argument : none

Return value : [out] none

Example

```
caoCtrl.Execute "ClearPacket"
```

3.7.6. CaoController::Execute ("SetTimeout") command

Sets a timeout period.

Syntax SetTimeout(<iTimeout>)

iTimeout : [in] Timeout period (msec) (VT_UI4)

Return value : [out] none

Example

```
caoCtrl.Execute "SetTimeout", 1000
```

3.7.7. CaoController::Execute ("GetTimeout") command

Gets the currently assigned timeout period

Syntax GetTimeout()

Argument : [in] none

Return value : [out] Timeout value (msec) (VT_UI4)

Example

```
Dim timeout as Long
timeout = caoCtrl.Execute("GetTimeout")
```

3.7.8. CaoController::Execute ("GetCommandResult") command

Waits for the completion of the asynchronous command to get the return value of it.

If the executed asynchronous command which has not return value (e.g,; ChangeModeAsync) is executed, it returns nothing.

If any synchronous command is used before this command, "Get result error" (0x80100003) occurs and no value will be returned.

If an asynchronous command, which is the target of GetCommandResult command, ends with an error, the error will be ignored within the process of asynchronous command, and the error occurs at GetCommandResult command execution.

If there is no response within the specified timeout-period during the waiting time of the asynchronous command completion, a time-out error (0x80000900) will occur.

Note that if another command is executed after an asynchronous command, the execution result of the asynchronous command that you've just get will be deleted.

Syntax [<vntRet> =]GetCommandResult ()

Argument : none

vntRet : [out] Return value of asynchronous command (VT_VARIANT)

The following shows how to obtain the return value of asynchronous inspection

Example

```
Dim vntResult as Variant
Call caoCtrl.Execute("ExecuteCommandAsync", "RO")
vntResult = caoCtrl.Execute("GetCommandResult")
```