V-Works for XG provider
KEYENCE XG-7000 series

Version 1.0.4

User's guide

July 01, 2019

[Remarks]

It is necessary to install "V-Works for XG ActiveX control" for XG-8000/7000 series being offered by the KEYENCE CORPORATION. to use the VWXG provider with PC.

[Revision history]

Version	Date	Content
1.0.0	2011-12-08 2012-07-17	First edition. Document versioning rules was changed.
1.0.1	2012-11-15	Added command. "TriggerAndGetResult".
1.0.2	2014-02-12	Added command. " ReadInspectSetting". Added option "Timeout".
1.0.3	2016-01-13	Added Asynchronous command. Added GetCommandResult command. Added RecievePacket command. Modified TriggerAndGetResult command.
1.0.3	2018-01-26	Note postscript of V-Works for XG.
1.0.4	2019-07-01	Modified a bug occurred at the disconnection of the provider.

Contents

1. Introduction

This book is a user's guide of the V-Works for XG provider (henceforth VWXG provider) that is the CAO provider of the V-Works for XG ActiveX control that operates in conjunction with the XG-7000 series vision system manufactured by KEYENCE. It is necessary to install "V-Works for XG ActiveX control" for XG-8000/7000 series being offered by the KEYENCE CORPORATION. to use the VWXG provider with PC.

VWXG provider notifies coordinated processing and the processing result the XG-7000 series controller and by way of the communication with which Ethernet is connected.

This book explains the function of this VWXG provider and the installation method.

2. Outline of provider

2.1. Outline

The VWXG provider offers the method by CaoController::Execute as a command execution method. CaoController::Execute can send and receive the command making use of the serial interface. [1]

Table2-1　VWXG provider

File name	CaoProvVWXG.dll
ProgID	CaoProv.KEYENCE.VWXG
Registry registration	regsvr32 CaoProvVWXG.dll
Deregistration	regsvr32 /u CaoProvVWXG.dll

2.2. Method and property

2.2.1. CaoWorkspace::AddController method

VWXG provider refers to the connection parameter for communication and connects the communication at the time of AddController. Option specifies a communication form at that time. [2]

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

bstrCtrlName	: [in] The controller name　arbitrary.
bstrProvName	: [in] Provider name (fixed to" CaoProv.KEYENCE.VWXG")
bstrPcName	: [in] Execution machine name of provider
bstrOption	: [in] Option character string

The following table is a list of option string items.

Table2-2Option character string of CaoWorkspace::AddController

Option	Meaning
Conn =< connection parameter >	Mandatory. Set the communication form and connection parameters. Please refer to 2.2.1.1 for details.

[1] In case of installation from ORiN2 SDK, register and unregister operation is not required.

[2] If AddController is executed in the environment in which V-Works for XG is not installed, the execution environment becomes unstable.

Timeout=[<Time-out time >]	Specify the communication time-out time (default: 500) msec.

2.2.1.1. Conn option

Connection parameter character string of the Conn option is shown as follows. The object in the square brackets ("[ ]") is omittable. The underlined part written in the each explanation below shows the default value in the event of without specifying options.

・ Ethernet device

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

<IP Address>	: IP address of connected XG-7000 series. Example:"192.168.0.10", "192.168.0.11"
<Port No>	: Connected port number. 8502,8503. . . Able to set arbitrarily.

2.2.2. CaoController::Execute method

Send and receive commands via Ethernet by No control sequence method. The command name is specified in the first argument and the parameter of the command is specified for the second argument. Please refer to Chapters 3 Command reference for details of each command.

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

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

2.2.3. Error code

When Execute method is performed, the processing result from the XG-7000 series is returned as HRESULT.

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

When the process completed with errors (ER) : 0x80100000 + return value

Exmple : ChangeMode is executed

hr = 0x801003EA : The value of argument is out of range.

For details about error contents, please refer to the Reference manual of V-Works for XG ActiveX control for XG-7000 series.

Table 2-3 Error code list

Error	Error number	Description
E_BASED_VWXG	0x80100000 | VWXG error	VWXG series original error
E_ENABLED_CANCEL	0x80101000	Failed to perform cancel process
E_EXECUTING	0x80102000	Another command was executed during a command execution
E_GET_COMMAND_RESULT	0x80103000	GetCommandResultcommand was executed after a Synchronous command execution

2.2.4. CaoController::OnMessage event

When the event of the XG-7000 series is issued, OnMessage event of CAO is issued.

Message:: Store the type of the event in the Number property

Message:: Store the detail of event in the Value property.

Please refer to the reference manual of the V-Works for XG ActiveX control for the XG-7000 series for details of each event of the XG-7000 series.

Table2-3 Event type list

Type	Explanation (XG-7000 series event)	Content of Value
0x000	The remote desktop screen update (OnRemoteDesktopUpdated)	(<iState>) < IState >= VT_I2: Contains the processing result 0 :Processing succeeded 1100: Communication exception
0x001	Receive results data output (OnResultLogDataReceived)	(<iState>, <iDriveNo>, <iSettingNo>, <iUnitId>, <bstrDstFile>) < IState >= VT_I2: Processing result 0 :Processing succeeded 1001: Processing failed 1004: Out of space 1100: Communication exception <iDriveNo >= VT_I2: Contains the SD-card number 1:SD1 2:SD2 Set with -1 if retrieval fails

		< ISettingNo >= VT_I2: Current inspection setting No (0～999) Set with -1 if retrieval fails <unit ID ..iUnitId >= VT_I2.. Set with the unit ID that provided the data output (0～999)　Set with -1 if retrieval fails. < BstrDstFile >= VT_BSTR: Contains the full path to the destination file (Contains null (empty) if the method fails)
0x002	Stop result logging (OnResultLogStopped)	(<iState>) < IState >= VT_I2: Processing result 0 :Processing succeeded 1100: Communication exception
0x00A	Execute no protocol command (OnCommandFinished)	(<iState>, <iKind>) < IState >= VT_I2: Processing result 0 :Processing succeeded < IKind >= VT_I2: Contains the type of command that generated this response For details on errors of <iState>, please refer to the item of "Event result" of each command of the V-Works control reference manual for the XG-7000 series. Please refer to the V-Works for XG ActiveX control reference manual for the XG-7000 series for the command list that responds.

3. Command reference

This chapter explains each command of the CaoController::Execute method. For detailed operation of each command, please refer to API list in the reference manual of the V-Works for XG ActiveX control for the XG-7000 series of KEYENCE.

Table3-1 CaoController::Execute command list

XG-7000 series command	Command	Function	
Remote desktop command			
StartRemoteDesktop	StartRemoteDesktop	Start the remote desktop.	P. 11
StopRemoteDesktop	StopRemoteDesktop	Stop the remote desktop.	P. 11
UpdateRemoteDesktop	UpdateRemoteDesktop	Update the remote desktop.	P. 12
CaptureRemoteDesktop	CaptureRemoteDesktop	Save an image of the remote desktop to a file.	P. 12
ClearRemoteDesktop	ClearRemoteDesktop	Clear the remote desktop screen.	P. 12
Result logging command			
StartResultLog	StartResultLog	Start result logging.	P. 13
StopResultLog	StopResultLog	Stop result logging.	P. 13
SetResultLogUnit	SetResultLogUnit	Specify the target unit for acquiring result data output.	P. 14
GetResultLogData	GetResultLogData	Retrieve results data output.	P. 14
No protocol command			
ExecuteCommand	ExecuteCommand	Execute Non-procedure command.	P. 15
	ExecuteCommandAsync	Execute Non-procedure command (Asynchronous)	P. 15
ChangeMode	ChangeMode	Change between Online/Offline mode.	P. 15
	ChangeModeAsync	Change between Online/Offline mode (Asynchronous)	P. 16
ReadMode	ReadMode	Read the current operating mode, Run or Offline.	P. 17
Reset	Reset	Issue a reset signal.	P. 17
Restart	Restart	Return to beginning of flow.	P. 18
	RestartAsync	Return to the beginning of flow (Asynchronous)	P. 18
Trigger	Trigger	Issue a trigger	P. 19
EnableTrigger	EnableTrigger	Enable/disable trigger input.	P. 19
ReadTriggerEnable	ReadTriggerEnable	Read trigger input enable status.	P. 19
WriteVariable	WriteVariable	Write variable values.	P. 20
ReadVariable	ReadVariable	Read the variable value	P. 20

ChangeInspectSetting	ChangeInspectSetting	Change the inspection setting file number.	P. 21
	ChangeInspectSettingAsync	Change the inspection setting file number (Asynchronous)	P. 21
ReadInspectSetting	ReadInspectSetting	Get the inspection setting file number.	P. 22
ClearError	ClearError	Clears system errors.	P. 22
Original enhancing command			
—	TriggerAndGetResult	Issue a trigger and get result.	P. 23
—	GetCommandResult	Get the return value of asynchronous command	P. 23
—	RecievePacket	Receive packet	P. 24

3.1. Remote, desktop command

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

Start the Remote Desktop function.

If Auto Update Mode has been specified, the OnRemoteDesktopUpDated event is issued each time the screen update completes. No screen data is transferred if the Manual Update Mode is specified. Transfer the screen data by calling the UpdateRemoteDesktop method.

Syntax StartRemoteDesktop(< lMode >)

< lMode >	:	The operational mode of a remote, desktop function is specified (VT_I4). 0:Auto Update Mode 1:Manual Update Mode
Return value	:	None

The following example shows how to start the Remote Desktop function by Auto Update Mode.

Example

```
Dim lMode as long
lMode = 0

caoCtrl.Execute "StartRemoteDesktop", lMode
```

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

Stop the Remote Desktop function.

Syntax StopRemoteDesktop( )

Argument	:	None
Return value	:	None

The following example shows how to stop the Remote Desktop function.

Example

```
        caoCtrl.Execute "StopRemoteDesktop"
```

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

Retrieve the monitor screen of the controller and updates the Remote Desktop screen.

The OnRemoteDesktopUpdated event is issued when the screen update completes.

Syntax UpdateRemoteDesktop( )

Argument	:	None
Return value	:	None

The following example shows how to update the Remote Desktop screen.

Example

```
        caoCtrl.Execute "UpdateRemoteDesktop"
```

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

Save the contents of the currently displayed Remote Desktop as a bitmap image file..

Syntax CaptureRemoteDesktop(< bstrFile >)

bstrFile	:	Specify the full path to the destination for the bitmap image file. (VT_BSTR)
Return value	:	None

The following example shows how to save the Remote Desktop screen in "D:¥Temp¥Img.bmp" file.

Example

```
        Dim bstrFile as string
        bstrFile = "D:¥Temp¥Img.bmp"

        caoCtrl.Execute "CaptureRemoteDesktop", bstrFile
```

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

Initialize the currently displayed Remote Desktop screen (to a black screen).

Syntax ClearRemoteDesktop( )

Argument	:	None
Return value	:	None

The following example shows how to initialize the displayed Remote Desktop screen.

Example

```
        caoCtrl.Execute "ClearRemoteDesktop"
```

3.2. Result logging command

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

Start the result logging function.

OnResultLogDataReceived event will notify you when results data output has been received.

Syntax StartResultLog(< iMode >, < bstrFolder >)

< iMode >	:	[in] Specify whether to save the received data to a file. (VT_I2) 0 : Save received data to a file. 1 : Do not save received data to a file.
< bstrFolder >	:	[in] Specify the full path to the destination for the received data file. (255 characters or less). This is only referenced if iMode parameter is 0. Folders SD1 and SD2 are created under this base folder, and the file is created according to the file naming rule. (VT_BSTR)

The following example shows how to start the result logging setting "C:¥work" as a destination for the received data

Example

```
        Dim bstrFolder as string
        bstrFolder = "C:¥work"

        caoCtrl.Execute "StartResultLog", Array(0, bstrFolder)
```

3.2.2. CaoController::Execute ("StopResultLog") command

Stop the result logging function.

The command only posts a request. The OnResultLogStopped event issues when result logging has stopped..

Syntax StopResultLog( )

Argument	:	None
Return value	:	None

The following example shows how to stop result logging.

Example

```
caoCtrl.Execute "StopResultLog"
```

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

Specify the Data output unit that will serve as the target for GetResultLogData method calls, and sets aside the most recent data for retrieval.

Syntax SetResultLogUnit(< iUnitId >)

< iUnitId >	:	Specify the Data output unit ID (0-999) (VT_I2).
Return value	:	None

The following example shows how to specify the unit number 5 as the target for acquiring the latest data.

Example

```
Dim iUnitId as integer
iUnitId = 5

caoCtrl.Execute "SetResultLogUnit", iUnitId
```

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

Retrieve the most recently received results data. This method retrieves data as text strings and numerical data specified in the Data output unit byspecifying four indexes.

Syntax [vntRet =] GetResultLogData(< iItemNo >, < iArrayNo >, < iMemberNo >)

< iItemNo >	:	Specify the output item number in the Data output unit (0-255) (VT_I2).
< iArrayNo >	:	Specify the array number in the output items. (Specify the position from the first array starting at 0.) Specify 0 if no arrays were set in the output items. (VT_I2).
< iMemberNo >	:	Specify the member number in the position, circle, or line type variables (0-2). Specify 0 for a scalar type variable. (VT_I2).

		0 :Positional type: X, circle type: CX, line type: T
		1 :Positional type: Y, circle type: CY, line type: RH
		2 :Circle type: CR
< vntRet >	:	Contains the numeric data of a string character formatted as specified in the Data output unit and output data value. (VT_VARIANT).

The following example shows how to acquire 0 (scalar variable) values of the output items.

Example

```
Dim iItemNo as integer
Dim iArrayNo as integer
Dim iMemberNo as integer
Dim vntResult as variant
iItemNo = 0
iArrayNo = 0
iMemberNo = 0

vntResult = caoCtrl.Execute "GetResultLogData", Array(iItemNo, iArrayNo, iMemberNo)
```

3.3. No protocol command

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

Execute the specified no protocol command. This method always receives a response whether the command executes successfully or not.

Please refer to the V-Works for XG ActiveX control reference manual for the XG-7000 series for the supported teletype protocol command.

Syntax [<vntRet> =]ExecuteCommand(< bstrCommand >)

bstrCommand	:	Specify the command as a string (VT_BSTR).
vntRet	:	Contains the command response. Contains null (empty) if the method fails.(VT_BSTR).

The following example shows how to specify No protocol command and change the XG-7000 state to online run mode.

Example

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("ExecuteCommand", "R0")
```

3.3.2. CaoController::Execute ("ExecuteCommandAsync") command

Execute a specified non-procedure command asynchronously.

To obtain the return value or the execution result of this command, use GetCommandResult command.

For information about supported non-procedure commands, please refer to the V-Works for XG ActiveX control reference manual for XG-7000 series.

To obtain and ccheck the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.3.19.CaoController::Execute ("GetCommandResult") command.

Syntax ExecuteCommandAsync(< bstrCommand >)

bstrCommand	:	Specify the character string of the command (VT_BSTR)
Return value	:	none

Executing the following example will specify Non-procedure command and switch XG-7000 to RUN mode.

Example

```
Dim vntResult as variant

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

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

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

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

Switch to online run mode or offline mode.

Syntax ChangeMode(< iMode >)

iMode	:	[in] Specify the desired mode.
		0 : Offline mode
		1 :Online run mode
Return value	:	None

The following example shows how to change the XG-7000 state to online run mode.

Example

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

3.3.4. CaoController::Execute ("ChangeModeAsync") command

Change the operation mode between Online/Offline mode asynchronously.

To obtain and ccheck the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.3.19.CaoController::Execute ("GetCommandResult") command.

Syntax ChangeModeAsync(< iMode >)

iMode	:	[in] Specify a desired operation mode
		0 : Offline mode
		1 : Online mode
Return value	:	none

Executing the following example will switch XG-7000 to Online mode.

Example

```
Dim vntResult as variant

caoCtrl.Execute "ChangeModeAsync", 1
vntResult = caoCtrl.Execute("GetCommandResult")

' vntResult : Empty
```

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

Retrieve the current operating mode (Online run mode, Offline mode, Remote Capture mode).

Syntax [< vntRet > =] ReadMode( )

Argument	:	None
< vntRet >	:	[out] Contains the current operating mode. Set to -1 if retrieval fails. (VT_I2)
		0 :Offline mode
		1 :Run mode
		2 :Remote capture mode

The following example shows how to retrieve the current operation mode of the XG-7000(Run mode) .

Example

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("ReadMode")

' vntResult : 1
```

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

Reset the controller.

Syntax Reset( )

Argument	:	None
Return value	:	None

The following example shows how to reset the controller.

Example

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

3.3.7. CaoController::Execute ("Restart") command

Jump to the next unit after the Start unit.

Syntax Restart( )

Argument	:	None
Return value	:	None

The following example shows how to use the Restart command..

Example

```
caoCtrl.Execute "Restart"
```

3.3.8. CaoController::Execute ("RestartAsync") command

Jump to the next unit after the Start unit asynchronously.

To obtain and ccheck the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.3.19.CaoController::Execute ("GetCommandResult") command.

Syntax RestartAsync( )

Argument	:	none
Return value	:	none

The following example shows how to use the RestartAsync command.

Example

```
Dim vntResult as variant

caoCtrl.Execute "RestartAsync"

‘ Obtain the return value of RestartAsync command
vntResult = caoCtrl.Execute("GetCommandResult")

‘ vntResult : Return value(Empty)
```

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

Issue a trigger.

Syntax Trigger(< iTriggerNo >)

iTriggerNo	:	Specify the number of the trigger to issue. (VT_I2) 1-4: Trigger 1-4 -1 :All triggers
Return value	:	None

The following example shows how to issue the trigger number 1.

Example

```
Dim iTriggerNo as integer
iTriggerNo = 1

caoCtrl.Execute "Trigger", iTriggerNo
```

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

Enable/disable trigger input.

Syntax EnableTrigger(< iMode >)

iMode	:	[in] Specify whether trigger is to be enabled or disabled. (VT_I2) 0 :Trigger disabled 1 :Trigger enabled
Return value	:	None

The following example shows how to disable the trigger input.

Example

```
Dim iMode as integer
iMode = 0

caoCtrl.Execute "EnableTrigger", iMode
```

3.3.11. CaoController::Execute ("ReadTriggerEnable") command

Retrieve current enable/disable status of trigger.

Syntax [< vntRet > =] ReadTriggerEnable( )

Argument	:	None
vntRet	:	Contains the trigger enable/disable status. Set to -1 if retrieval fails.

	(VT_I2)
0	:Trigger is disabled
1	:Trigger is enabled

The following example shows how to retrieve the trigger status of the XG-7000(Trigger is enabled).

Example

```
Dim vntResult as variant

vntResult = caoCtrl.Execute "ReadTriggerEnable"
```

3.3.12. CaoController::Execute ("WriteVariable") command

Write a value to a specified scalar type variable (global or local).

Syntax WriteVariable(< bstrName >, < dblValue >, < iSyncMode >)

< bstrName >	:	Specify the scalar variable name as a string. (VT_BSTR)
< dblValue >	:	Specify the value to write the variable. (VT_R8)
< iSyncMode >	:	Specify how the value should be synchronized and reflected to the flowchart. (VT_I2)
Return value	:	None

The following example shows how to write the value to a scalar type variable "#MyVar".

Example

```
Dim bstrName as string
Dim dblValue as double
bstrName = "#MyVar"
dblValue = 100

caoCtrl.Execute "WriteVariable", Array(bstrName, dblValue, 0)
```

3.3.13. CaoController::Execute ("ReadVariable") command

Retrieve the value of a specified scalar variable type variable.

Syntax [< vntRet > =] ReadVariable(< bstrName >)

bstrName	:	Specify the scalar variable name as a string. (VT_BSTR)
vntRet	:	Contains the variable value. Set to -1.0 if retrieval fails. (VT_R8)

The following example shows how to retrieve the value of the scalar type variable "#MyVal(=100)".

Example

```
Dim bstrName as string
Dim dblValue as double
bstrName = "#MyVar"

dblValue = caoCtrl.Execute "ReadVariable", bstrName
```

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

Switch the current program to the specified inspection setting number and SD card.

Syntax ChangeInspectSetting (< iDriveNo >, < iSettingNo >)

iDriveNo	:	Specify the SD card that the setting number is stored on. (VT_I4)
		1 : SD1
		2 : SD2
iSettingNo	:	Specify the inspection setting number. (0～999)(VT_I4)
Return value	:	None

The following example shows how to change the inspection setting number of SD card (SD1) to No.1.

Example

```
Dim iDriveNo as integer
Dim iSettingNo as integer
iDriveNo = 1
iSettingNo = 1

Call caoCtrl.Execute("ChangeInspectSetting", Array(iDriveNo, iSettingNo))
```

3.3.15. CaoController::Execute ("ChangeInspectSettingAsync") command

Switch the current program to the specified inspection setting number and SD card asynchronously.

To obtain and ccheck the return value of the command, use GetCommandResult command. For details about GetCommandResult, please refer to 3.3.19.CaoController::Execute ("GetCommandResult") command.

Syntax [< vntRet > =] ChangeInspectSettingAsync (< iDriveNo >, < iSettingNo >)

iDriveNo	:	Specify the SD card that the setting number is stored on (VT_I4)
		1 : SD1
		2 : SD2
iSettingNo	:	Specify the inspection setting number. (0 to 999)(VT_I4)
Return value	:	none

The following example shows how to change the inspection setting number of SD card (SD1) to No.1

Example

```
Dim vntResult as variant
Dim iDriveNo as integer
Dim iSettingNo as integer
iDriveNo = 1
iSettingNo = 1

Call caoCtrl.Execute("ChangeInspectSettingAsync", Array(iDriveNo, iSettingNo))

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

' vntResult : Return value(Empty)
```

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

Get the current program to the specified inspection setting number and SD card.

Syntax [< vntRet > =] ChangeInspectSetting ()

Argument	:	[in] None
Return Value	:	[out] parameters (< iDriveNo >,<iSettingNo >) (VT_I4|VT_ARRAY)
		iDriveNo: Specify the SD card that the setting number is stored on.
		iSettingNo: Specify the inspection setting number.

Following example shows.

Example

```
Dim vntRet as Variant

vntRet = caoCtrl.Execute("ReadInspectSetting")
' vntRet(0) : DriveNo
' vntRet(1) : SettingNo
```

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

Clear the specified error status type and error code.

Syntax ClearError(< iErrorNo >)

iErrorNo	:	Set the error status type. (VT_I2)
		0 : Clear %Error0 and %Error0Code.
		1 : Clear %Error1 and %Error1Code.
Return value	:	None

The following example shows how to clear the error that error number is 0.

Example

```
Dim iErrorNo as integer
iErrorNo = 0

caoCtrl.Execute "ClearError", iErrorNo
```

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

. Issue a trigger and get output data.

Syntax TriggerAndGetResult (< iTriggerNo >)

iTriggerNo	:	Specify the number of the trigger to issue. (VT_I2) 1-4: Trigger 1-4 -1 :All triggers
Return value	:	Get outputdata. (BT_BSTR)

The following example shows how to get outputdata.

Example

```
Dim iTrigger as integer
Dim vntRet as Variant

iTrigger = -1

vntRet = caoCtrl.Execute("TriggerAndGetResult", iTrigger)
```

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

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

If the executed asynchronous command which has not return value 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 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
caoCtrl.Execute "ReStartAsync"
vntResult = caoCtrl.Execute("GetCommandResult")
```

3.3.20. CaoController::Execute ("ReceivePacket") command

Receive packets.

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

Syntax <vntRet> = RecievePacket ()

Argument	:	none
vntRet	:	[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

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