In-Sight provider
Cognex

Version 1.0.5

User's guide

March 20, 2024

[Remarks]

[Revision history]

Version	Date	Content
1.0.0	2011-11-24 2012-07-17	First edition. Document versioning rules was changed.
1.0.1	2012-09-11	SendMessageAndWait was changed. Add GetMessage command.
1.0.1	2012-09-27	Correct GetFileList command.
1.0.2	2013-01-09	Add commands. SendMessageAndGetEZ, SendMessageEZ, GetEZ, SetTimeoutNM, GetTimeoutNM Change command. AddController
1.0.3	2013-03-19	Add ResetEZ command.
	2015-09-09	Add an error code.
1.0.4	2016-02-25	Modification of GetFilelist command. Add ExecuteCommand command. Add RecievePacket command.
1.0.5	2024-03-20	Improve stability during In-Sight connection.

Contents

1. Introduction

This book is a user's guide of the In-Sight provider that is the CAO provider for the vision system manufactured by Cognex.

The In-Sight provider connects with the Ethernet-connected In-Sight series by means of Native Mode and the DataChannel mode to send and receive the command of Native Mode or to get images. The communication supports the Ethernet (TCP/IP) connection. Native Mode does not support the XML format output, so please invalidate the XML format when using this provider.

2. Outline of provider

2.1. Outline

The In-Sight provider offers two command execution methods by CaoController::Execute and CaoVariable.

CaoController::Execute can send and receive In-Sight and the command according to Native Mode communication.

CaoVariable executes In-Sight value acquisition and acquire images via Native Mode communication and part of the DataChannel communication.

2.2. Method and property

2.2.1. CaoWorkspace::AddController method

In-Sight provider refers to the connection parameter for communication and connects the communication at the time of AddController. Option specifies communication form, time-out time, user name, password and the port number of the Telnet.

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

bstrCtrlName : [in] Controller name arbitrary.
bstrProvName : [in] Provider name. (Fixed to "CaoProv.Cognex.In-Sight")
bstrPcName : [in] Execution machine name of provider
bstrOption : [in] Option character string

Following is a list of option string items.

Table 2-1 Option string of CaoWorkspace::AddController

Option	Description
Conn =< connection parameter >	Mandatory. Set the communication form and connection parameters.
User=[<Name of user>]	Specify user names to log in In-Sight. (Default: Admin)
Password[=<Password>]	Specify password to log in. (Default: Not specified)
Timeout[=<Timeout period>]	Specify time-out time (msec) for sending and receiving. (Default: 500)
EZPort[=<Port Number>]	Specify the port number to receive the setting results of EasyBuilder. If this option is omitted, the setting result of the EasyBuilder cannot

	be received. For information of setting methods, refer to 4.2.

2.2.1.1. Conn option

Following is connection parameter string for Conn option.

- **Ethernet device**

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

  <IP Address> : : Mandatory. Specify IP address.
  Example:"127.0.0.1"

  <Port No> : : Specify the port number to communicate Native Mode.
  (Default: 23)
  Example:"127.0.0.1:23"

2.2.2. CaoController::Execute method

Send and receive commands of Native Mode. The command name is specified in the first argument and the parameter of the command is specified for the second argument. Please refer to Chapter 3 Command reference for details of each command.

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

bstrCommandName: [in] Command name

vntParam : [in] Parameter

At the time of the Execute method execution, the return value of the status code from In-Sight is returned as HRESULT.

Return value 1: S_OK (0)

Return value 0 ～ -9: 0x80100010 + abs (return value)

Example: SetEvent execution

hr = 0x80100012: Fail to command execution or system is off-line.

Please refer to In-Sight Explorer Reference of Cognex for the content of the error.

2.2.3. CaoController::AddVariable method

Create variables used for acquiring images and values of cells. Please refer to Table 2-2 Controller class system variable list for the acquisition of images. When acquire the value of cell, please specify the variable name as arbitrary character strings, and write the cell position in the option character strings.

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

bstrVariableName : [in] Variable name

bstrOption : [in] Option character strings (Value of Cell = Cell number, only when acquiring cell value).

Example

```
Dim objA002 as Object
Dim bstrA002 as String
objA002 = caoCtrl.AddVariable("A002", "Cell=A002")
bstrA002 = objA002.Value

bstrA002: "1.000"
```

2.2.4. CaoVariable::put_value property

Variable class currently does not support put_value property.

2.2.5. CaoVariable::get_value property

The image can be acquired in the format of Table 2-2 Controller class system variable list.

The value of the cell is acquired with VT_BSTR.

2.3. Variable list

2.3.1. Controller class

Table 2-2 Controller class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@BITMAP	VT_UI1|VT_ARRAY	Get image in the BITMAP format via the Native Mode. Attention: AddVariable command cannot be done unless camera images are acquired by In-Sight once.	✓	-
@BITMAP_DC	VT_UI1|VT_ARRAY	Get camera images in BITMAP format via DataChannel. To acquire the camera image, camera need to be updated with the trigger. The following option character strings can be used. Screen=1 (Default), 2, 4: Decide the size of the acquiring image. The size is 1/Screen.	✓	-

		Port=5000 (Default): Specify the port number of DataChannel. Timeout=500 (Default): Set the time-out of DataChannel SM8=False (Default): When execute GetValue command, transmit SM8 and active trigger ON.		
@RAWIMG	VT_UI1|VT_ARRAY	Get images through Native Mode in RAW image format. * Unimplemented	✓	-
@RAWIMG_DC	VT_UI1|VT_ARRAY	Get the camera image through DataChannel in RAW image format. * Unimplemented	✓	-

2.4. Error code of In-Sight provider

In the In-Sight provider, specific error codes shown below are designated. About the ORiN2 commonness error, please refer to the chapter of the error code of "ORiN2 Programming guide".

Table 2-3 Custom Error

Error name	Error code	Explanation
E_INSIGHTERROR	0x80100010 ～ 0x80100019	In-Sight Error (Refer to 2.2.2)
E_INVALIDPACKET	0x80100020	Received invalid packet
E_CONNECTION	0x80100021	Communication disconnected
E_INVALIDPASSWORD	0x80100022	Invalid user name / password
E_UNPREPAREDFORIMAGE	0x80100023	Preparation for image acquisition is not finished.
E_EZERROR	0x80100024	A port for result acquisition of EasyBuilder is not specified.

3. Command reference

This chapter explains each command of the CaoController::Execute method. For detailed operation of each command, please refer to the communication reference of In-Sight Explorer Reference made by Cognex.

Table 3-1 CaoController::Execute command list

In-Sight Command	Command	Function	
Basic communication			
-	NativeMode	Control the In-Sight device by the Telnet communication.	P. 12
File command and job command			
Load File	LoadFile	The job that specifies it is loaded from the memory, and it makes it to an active job.	P. 12
Store File	StoreFile	Stores an active job in the memory.	P. 13
Read File	ReadFile	Unimplemented	
Write File	WriteFile	Unimplemented	
Delete File	DeleteFile	Delete the job from the memory.	P. 13
Get File	GetFile	Get the file name of an active job on the In-Sight device.	P. 13
Set Job	SetJob	Load a job from the job slot in the memory on the In-Sight processor, and be the jobs as an active job.	P. 14
Store Job	StoreJob	Store the current job in the specified job slot in the memory on the In-Sight processor.	P. 14
Read Job	ReadJob	Unimplemented	
Write Job	WriteJob	Unimplemented	
Delete Job	DeleteJob	Delete the job from the specified job slot in the memory on the In-Sight processor.	P. 14
Get Job	GetJob	Return the job slot of In-Sight into which an active job is loaded.	P. 15
File command and job command			
Read BMP	ReadBMP	Unimplemented	
Read Image	ReadImage	Unimplemented	
Write BMP	WriteBMP	Unimplemented	
Write Image	WriteImage	Unimplemented	
Set command and cell value command			
Get Value	GetValue	Return the value included in the specified cell.	P. 15
Set Integer	SetInteger	Set the edit box control stored in the cell to the specified integral value.	P. 16
Set Float	SetFloat	Set the edit box control stored in the cell to the specified floating point value.	P. 16
Set String	SetString	Set the edit box control stored in the cell to the specified character string.	P. 17

Get Info	GetInfo	Return system information from the In-Sight device.	P. 17
Read Settings	ReadSettings	Unimplemented	
Write Settings	WriteSettings	Unimplemented	
Store Settings	StoreSettings	Store the current setting of the In-Sight device in the proc.set file in the memory.	P. 17
Set IP Lock	SetIPLock	Prevent IP address tampering of the In-Sight device.	P. 18
Get IP Lock	GetIPLock	Return the state of security of IP address (accessible/not accessible)	P. 18
Execution command and online command			
Set Online	SetOnline	Set the In-Sight device into online or offline mode.	P. 19
Get Online	GetOnline	Return the online state of the In-Sight processor.	P. 19
Set Event	SetEvent	Trigger a specified event (image acquisition, etc..).	P. 19
Set Event & Wait	SetEventAndWait	Trigger a specified event (image acquisition, etc.), and waits until the command is completed to return a response.	P. 20
Reset System	ResetSystem	Unimplemented	
Abort Execution	AortExecution	Unimplemented	
Send Message	SendMessage	Send ASCII character string to an In-Sight spread sheet over a Native Mode connection.	P. 20
Enhancing command of Native Mode			
Get Connections	GetConnections	Unimplemented	
Get Expr	GetExpr	Unimplemented	
Get Filelist	GetFilelist	Return the number of files stored in memory, and the name of each file in memory on the In-Sight device.	P. 21
Put Live	PutLive	Unimplemented	
Put Portnum	PutPortnum	Unimplemented	
Put Timeout	PutTimeout	Unimplemented	
Put Update	PutUpdate	Unimplemented	
Put Watch	PutWatch	Unimplemented	
Put XML	PutXML	Unimplemented	
Original enhancing command			
-	SendMessageAndWait	Software triggers ON and receives "WriteMessage".	P. 21
-	GetMessage	Receives "WriteMessage".	P. 22
-	SendMessageAndGetEZ	Software triggers ON and receives TCP communication output result which is set by EasyBuilder.	P. 23
-	SendMessageEZ	Software triggers ON and prepares to receive the results by GetEZ command.	P. 23
-	GetEZ	Receive the result which is set by EasyBuilder	P. 24
-	ResetEZ	Set a flag for GetEZ command.	P. 25
-	SetTimeoutNM	Set time-out period for communicating with In-Sight.	P. 25
-	GetTimeoutNM	Acquire time-out period for communicating with In-Sight.	P. 25
-	ExecuteCommand	Send raw data.	P. 26
-	RecievePacket	Receives raw data.	P. 26

3.1. Basic communication

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

Send and receive Native Mode command.

Syntax NativeMode (<bstrSyntax>, [<bstrOption>])

< bstrSyntax > : [in] Syntax (VT_BSTR)

<bstrOption> : [in] BSTR option (VT_BOOL)

False	Return the return value with VT_UI1 | VT_ARRAY. (Default)
True	Return the return value with VT_BSTR.

Return value : [out] Status code and return value of command (VT_UI1 | VT_ARRAY or VT_BSTR)

This command allows communicating with In-Sight by Native Mode. For "bstrSyntax", set the command name and the parameter by the character string.

Example

```
Dim vntResult as Variant
vntResult = caoCtrl.Execute("NativeMode", Array("GF", "BSTR=True"))

vntResult : "1
         MyJob1.job"
```

3.2. File command and job command

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

Load the specified job from the memory, making it the active job.

Attention: The In-Sight device should be off-line.

Syntax LoadFile (<bstrFileName>)

< bstrFileName > : [in] File name of the job (VT_BSTR)

Return value : None

Load the job that specified by "bstrFileName" from the memory on the In-Sight processor, making it the active job.

Example

```
caoCtrl.Execute "LoadFile", "MyJob1.job"
```

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

Save the current job in the memory on the In-Sight processor.

Attention: The In-Sight device should be off-line.

Syntax StoreFile (<bstrFileName>)

<bstrFileName> : [in] File name of the job (VT_BSTR)

Return value : None

Name the current job "bstrFileName", and saves it in the memory on the In-Sight processor.

Example

```
caoCtrl.Execute "StoreFile", "MyJob1.job"
```

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

Delete the job from the specified job slot in the memory on the In-Sight processor.

Attention: The In-Sight device must be off-line.

Syntax DeleteFile (< bstrFileName >)

< bstrFileName > : [in] File name of the job (VT_BSTR)

Return value : None

Delete the job that specifies by "bstrFileName".

Example

```
caoCtrl.Execute "DeleteFile", "MyJob1.job"
```

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

Return the file name of the active job on the In-Sight device.

Attention: The active job must be saved before this command can be executed successfully. If the job has been dragged and dropped, the file name of the job is returned.

Syntax GetFile ()

Argument : None

Return value : [out] bstrFilename(VT_BSTR)

Get the file name of the active job by VT_BSTR.

Example

```
Dim bstrFilename as String
bstrFilename = caoCtrl.Execute("GetFile")

bstrFilename : "MyJob1.job"
```

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

Load a job from the job slot in the memory on the In-Sight processor, making it the active job.

Attention: To use the job ID number feature, the job to be loaded must be saved with a numerical prefix of 0 to 999. For backwards compatibility on non-network capable In-Sight sensors, the prefix must be in the range of 0 to 19.

The In-Sight system must be off-line.

Syntax SetJob (<lID>)

<lID> : [in] Job ID(VT_I4) (0-999)

Return value : None

Make the IID-designate job as an active job.

Example

```
caoCtrl.Execute "SetJob", 2
Active job "2MyJob1.job"
```

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

Save the current job into the specified job slot in the memory on the In-Sight processor.

Syntax StoreJob (<lID>, <bstrJobName>)

<lID> : [in] Job ID (VT_I4) (0-19)

<bstrJobName> : [in] Job name (VT_BSTR)

Return value : None

Name the current job by "bstrJobName" and save it in the slot of lID.

Example

```
caoCtrl.Execute "StoreJob", Array(3, "TEST.job")
```

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

Delete the job from the specified job slot in the memory on the In-Sight processor.

Attention: For backwards compatibility, the job ID number must be in the range of 0 to 19.

The In-Sight device must be off-line.

Syntax DeleteJob (<lID>)

<lID> : [in] Job ID (VT_I4) (0-999)

Return value : None

Delete the job that specified by lID.

Example

```
caoCtrl.Execute "DeleteJob", 2
```

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

Return the ID of the active job on the In-Sight device.

Attention: To use the job ID number feature, the job to be loaded must be saved with a numerical prefix of 0 to 999. For backwards compatibility on non-network capable In-Sight sensors, the prefix must be in the range of 0 to 19.

The active job must be saved with a numerical prefix before this command can be executed successfully. If the job has been dragged and dropped, the file name must have a numerical prefix before this command can be executed successfully.

Syntax GetJob ()

Argument : None

Return value : [out] Job ID (VT_I4) (0-999)

Return the Active job ID by VT_I4.

Example

```
Dim lID as Long
lID = caoCtrl.Execute("GetJob")

lID : 2
```

3.3. Set command and cell value command

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

Return the value included in the specified cell.

Attention: An In-Sight cell containing any numeric value will return a float value formatted to 3 decimal places when requested by the GetValue command, regardless of whether the cell contains an integer or a floating-point value. If the cell contains a non-printing character, such as a Structure, a pound character (#) will replace the non-printing character. If the cell is empty, a null string will be sent.

Syntax GetValue (<bstrRow>, <lCol>)

<lCol> : [in] The column letter of the cell value to get (VT_BSTR) (A-Z)

<bstrRow> : [in] The row number of the cell value to get (VT_I4) (0-399)

Return value : [out] The value of cell (VT_BSTR)

Return the value of the cell specified by < bstrRow > and < lCol > by VT_BSTR.

Example

```
Dim bstrVal as String
bstrVal = caoCtrl.Execute("GetValue", Array("B", 1))

bstrVal: 1.000
```

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

Set the control contained in a cell to the specified integer value. The control must be of the types EditInt, Checkbox, or ListBox.

Syntax SetInteger(<bstrRow>, <lCol>, <lValue>)

<lCol> : [in] The column letter of the cell value to set (VT_BSTR) (A-Z)

<bstrRow> : [in] The row number of the cell value to set. (VT_I4) (0-399)

<lValue> : [in] The integer value to set. (VT_I4)

Return value : None

Set the value of the cell specifies by < bstrRow > and < lCol > to a value of < lValue >.

Example

```
caoCtrl.Execute "SetInteger", Array("B", 1, 100)
```

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

The edit box control stored in the cell is set to the specified floating point value. The edit box control should be EditFloat type.

Syntax SetFloat (<bstrRow>, <lCol>, <fValue>)

<lCol> : [in] The column letter of the cell value to set (VT_BSTR) (A-Z)

<bstrRow> : [in] The row number of the cell value to set (VT_I4) (0-399)

<fValue> : Value to set (VT_R4)

Return value : None

Set the value of the cell specifies by < bstrRow > and < lCol > to a value of < fValue >.

Example

```
caoCtrl.Execute "SetFloat", Array("B", 1, 10.130)
```

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

Set an edit box control contained in a cell to a specified string. The edit box must be of the type EditString.

Syntax SetString (<bstrRow>, <lCol>, <bstrValue>)

<lCol>	:	[in] The column letter of the cell value to set (VT_BSTR) (A-Z)
<bstrRow>	:	[in] The row number of the cell value to set. (VT_I4) (0-399)
<bstrValue>	:	The value to set (VT_BSTR)
Return value	:	None

Set the value of the cell that specifies by < bstrRow > and < lCol > to a value of < bstrValue >.

Example

```
caoCtrl.Execute "SetString", Array("B", 1, "Test")
```

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

Return the system information on the In-Sight device.

Syntax GetInfo ()

Argument	:	None
Return value	:	[out] System information (VT_BSTR | VT_ARRAY)

Get the system information on the In-Sight device. The result is stored in order of the serial number, the application version, the monitor version, the MAC address, and the build at the date.

Example

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

vntResult : "Serial Number: Z83560368",
      "Application Version: 4.05.00 (233)",
      "Monitor Version: 4.01",
      "MAC Address: 00-d0-24-02-41-a1",
      "Date of Build: Apr 4 2011, 13:18:09"
```

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

Store the In-Sight system setting to the proc.set file.

Syntax StoreSettings ()

Argument	:	None
Return value	:	None

Save the system settings.

Example

```
caoCtrl.Execute "StoreSettings"
```

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

Prevent unauthorized changes to an In-Sight device's IP address. When IP address is locked, a change of IP address which is done by the user whose access level is "Protected" or "Locked" by means of the "Camera connection manager" is not saved.

Attention: When IP address is locked, a user whose access level is "Protected" or "Locked" is able to log on the In-Sight device, but cannot change IP address.

Syntax StoreSettings (<lNum>)

<lNum> : [in] Parameter to be set (VT_I4)
0 : Unlock the IP address.
1 : Lock the IP address.

Return value : None

Set IP address state as specified by < lNum >.

Example

```
caoCtrl.Execute "SetIPLock", 1
```

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

Return the security status of the IP address on an In-Sight sensor. The ability to prevent unauthorized changes to the IP address is established by the SetIPLock command.

Syntax GetIPLock ()

Argument : None

Return value : [out] Security state of the IP address (VT_I4)
0 : The IP address is not locked.
1 : The IP address is locked.

Return the security state of the IP address by VT_I4.

Example

```
Dim lLock as long
lLock = caoCtrl.Execute("GetIPLock")
lLock : 1
```

3.4. Execution command and online command

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

Set the In-Sight device into online or off-line mode.

Attention: This command cannot place the In-Sight device into online mode if the device has been set Offline either manually (in the system tool bar) or by a Discrete Input signal before.

Syntax SetOnline (<lNum>)

<lNum> : [in] Online state of In-Sight device (VT_I4)
0 : Set the In-Sight device off-line.
1 : Set the In-Sight device online.

Return value : None

Set the online state of the In-Sight by lNum.

Example

```
caoCtrl.Execute "SetOnline", 1
```

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

Return the online state of the In-Sight processor.

Syntax GetOnline ()

Argument : None

Return value : [out] Online state of the In-Sight device (VT_I4)
0 : The In-Sight processor is currently off-line.
1 : The In-Sight processor is currently online.

Return the online state of the In-Sight processor in lNum.

Example

```
Dim lState as Long
lState = caoCtrl.Execute("GetOnline")

lState : 1
```

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

Trigger a specified event.

Attention: The In-Sight device must be online.

Syntax SetEvent (<lEventNum>)

<lEventNum> : [in] The event code to set (VT_I4)
0-7: Specifies a soft trigger.

8: Acquire an image and update the spread sheet. This option requires the AcquireImage function's Trigger parameter to be set to Camera, External, or Manual.

Return value : None

Execute software trigger on In-Sight.

Example

```
caoCtrl.Execute "SetEvent", 8
```

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

Triggers a specified event and waits until the command is completed to return a response.

Attention: The In-Sight device must be online.

E_TIMEOUT might be return before the command is completed when the Timeout setting time is short.

Syntax SetEventAndWait (<lEventNum>)

<lEventNum> : Event code in which in is set (VT_I4)

0-7: Specify a soft trigger.

8: Acquire an image and update the spreadsheet. This option requires the AcquireImage function's Trigger parameter to be set to Camera, External or Manual.

Attention: Camera trigger is not supported in the sensor of the In-Sight 1700 series, the In-Sight 1000 series or the In-Sight 4000 series.

Return value : None

Execute software trigger on In-Sight.

Example

```
caoCtrl.Execute "SetEventAndWait", 8
```

3.4.5. CaoController::Execute ("SendMessage") command

Sends ASCII string to an In-Sight spreadsheet over a Native Mode connection, and optionally, triggers a spreadsheet Event.

Syntax SendMessage (<bstrMessage>, [<lNum>])

< bstrMessage > : [in] Character string to set (VT_BSTR)

Attention: The string does not need to be enclosed with quotation marks in this command although it is required in the

command of In-Sight.

<lNum> [in] The event code to set (VT_I4)

0-7: Specify a soft trigger.

8: Acquire an image and update the spreadsheet. This option requires that the AcquireImage function's Trigger parameter be set to Camera, External or Manual.

Attention: This is an optional parameter.

Return value : None

Send BstrMessage to the In-Sight. The event trigger can be executed by specifying lNum.

Example

```
caoCtrl.Execute "SendMessage", Array("hoge", 8)
```

3.5. Extended Native Mode command

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

Return the name list of files in memory on the In-Sight device.

Attention: This command does not return the number of files stored in memory although it is returned in the command of In-Sight.

Syntax GetFilelist ()

Argument : None

Return value : [out] File list (VT_BSTR | VT_ARRAY)

Return file list stored in In-Sight to VT_BSTR | VT_ARRAY

Example

```
bstrFilenames = caoCtrl.Execute("GetFilelist")
bstrFilenames : array("TEST.job", "hoge.job")
```

3.6. Original extended command

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

After the SendMessage command is issued, receive the character string output by the WriteMessage function of In-Sight.

Attention: The In-Sight device must be online.

Syntax SendMessageAndWait ([<bstrMessage>], [<lNum>], [< lTerminateNum >])

<bstrMessage> : [in] Character string to set (VT_BSTR)

Attention: The character string does not need to be enclosed

with quotation marks in this command although it is required in the command of In-Sight. When omitting it, Null character is sent.

<lNum> [in] Event code to set (VT_I4)

0-7: Specify a soft trigger.

8: Acquire an image and update the spreadsheet. This option requires that the AcquireImage function's Trigger parameter be set to Camera, External or Manual.

Attention: When omitting it, eight is used.

< lTerminateNum> [in] terminator code to set (VT_I4)

0: None.

1: CR

2: LF

3: CR+LF

Attention: When omitting it, one is used.

Return value : [out] Text string to be output (VT_BSTR)

After the event is sent to In-Sight, receive the character string output by the WriteMessage function. When the WriteMessage function is not set, E_TIMEOUT is generated.

Example

```
bstrMessage = caoCtrl.Execute("SendMessageAndWait", Array("", 8, 1))
bstrFilenames : "SendMessage"
```

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

Receive the character string output by the WriteMessage function of In-Sight.

Attention: The In-Sight device must be online.

Syntax SendMessageAndWait ([< lTerminateNum>])

< lTerminateNum> [in] terminator code to set (VT_I4)

0: None.

1: CR

2: LF

3: CR+LF

Attention: When omitting it, one is used.

Return value : [out] Text string to be output (VT_BSTR)

Receive the character string output by the WriteMessage function. When the WriteMessage function is not set, E_TIMEOUT is generated.

Example

```
bstrMessage = caoCtrl.Execute("GetMessage",1)
bstrMessage : "SendMessage"
```

3.6.3. CaoController::Execute ("SendMessageAndGetEZ") command

After the SendMessage command is issued, receive the character string output by EasyBuilder communication (TCP/IP) of In-Sight. For information of EasyBuilder setting, refer to 4.2.

Attention : The In-Sight device must be online.

Syntax SendMessageAndGetEZ ([<bstrMessage>], [<lNum>], [<lTimeOut>])

<bstrMessage>	:	[in] Character string to set (VT_BSTR) Attention: The character string does not need to be enclosed with quotation marks in this command although it is required in the command of In-Sight. When omitting it, Null character is sent.
<lNum>		[in] The event code to set (VT_I4) 0～7 : Specifies a soft trigger. 8 : Acquire an image and update the spread sheet. This option requires the AcquireImage function's Trigger parameter to be set to Camera, External, or Manual. Attention: When omitting it, eight is used.
< lTimeOut >		[in] Maximum waiting time to receive. (msec) (VT_UI4) Attention: If this is omitted, wait 500 msec.
Return value	:	[out] Output text character strings. (VT_BSTR)

After sending an event to the In-Sight, receive the character string configured by EasyBuilder communication settings. E_TIMEOUT might be returned if the communication setting is not completed or, the time-out period is over. If EZPort option is not specified by AddController, an error of E_EZERROR will occur.

Example

```
bstrMessage = caoCtrl.Execute("SendMessageAndGetEZ", Array("", 8, 1000))
bstrFilenames : "SendMessage"
```

3.6.4. CaoController::Execute ("SendMessageEZ") command

After SendMessage command is issued, prepare to receive the character string output by EasyBuilder communication (TCP/IP) of In-Sight. To receive the character string, use GetEZ command.

Attention: The In-Sight device must be online.

Syntax SendMessageEZ ([<bstrMessage>], [<lNum>])

<bstrMessage> : [in] Character string to set (VT_BSTR)

Attention: The character string does not need to be enclosed with quotation marks in this command although it is required in the command of In-Sight. When omitting it, Null character is sent.

<lNum> [in] The event code to set (VT_I4)

0～7 : Specifies a soft trigger.

8 : Acquire an image and update the spread sheet.. This option requires the AcquireImage function's Trigger parameter to be set to Camera, External, or Manual.

Attention: When omitting it, eight is used.

Return value : None

GetMessage command can receive only the latest result of the SendMessageEZ command.

Example

```
Call caoCtrl.Execute("SendMessageEZ", Array("", 8))
```

3.6.5. CaoController::Execute ("GetEZ") command

Receive the character string output by EasyBuilder communication (TCP/IP) of In-Sight. When receiving the result by GetEZ command, activate trigger ON by SendMessageEZ command.

Attention: The In-Sight device must be online..

Syntax GetEZ ([<lTimeOut >])

< lTimeOut > : [in] Maximum waiting time to receive. (msec) (VT_UI4)

Attention: If this is omitted, wait 500 msec.

Return value : [out] Output text character strings. (VT_BSTR)

GetEZ command can receive the latest result of the SendMessageEZ command or use ResetEZ command. E_TIMEOUT might be return when the SendMessageEZ command is not running or, the result output is not configured by EasyBuilder. If EZPort option is not specified by AddController, an error of E_EZERROR will occur.

Example

```
bstrResult = caoCtrl.Execute("GetEZ", 1000)
```

3.6.6. CaoController::Execute ("ResetEZ") command

Set a flag for GetEZ command. You can receive the character string output by EasyBuilder communication (TCP/IP) of In-Sight only once by using GetEZ command.

Syntax ResetEZ

Argument : None

Return value : None

You must call this command before In-Sight sends the character string.

Example

```
Call caoCtrl.Execute("ResetEZ")
bstrResult = caoCtrl.Execute("GetEZ", 1000)
```

3.6.7. CaoController::Execute ("SetTimeoutNM") command

Change the time-out period at the time of communicating with Native Mode. Initial value is the value set by Option (Timeout) of AddController.

Syntax SetTimeoutNM (<lTimeOut >)

< lTimeOut > : [in] Timeout period to set (msec) (VT_UI4)

Return value : None

Example

```
Call caoCtrl.Execute("SetTimeoutNM", 1000)
```

3.6.8. CaoController::Execute ("GetTimeoutNM") command

Acquire time-out period for communicating with Native Mode.

Syntax GetTimeoutNM ()

Argument : None

Return value : [out] Configured time-out period. (msec) (VT_UI4)

Example

```
lTimeout = caoCtrl.Execute("GetTimeoutNM")
```

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

Transmits raw data.

Syntax ExecuteCommand (< Raw Data >)

< Raw Data > : [in] Raw data to be sent (VT_BSTR)

Return value : None

Example

```
Call caoCtrl.Execute("ExecuteCommand", "GetTimeoutNM")
```

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

Receives raw data.

Syntax RecievePacket ()

Argument : None

Return value : [out] Received raw data (VT_BSTR)

Example

```
strRowReciveData = caoCtrl.Execute("RecievePacket")
```

4. How to acquire image processing results

There are two ways to get results from In-Sight.

1. Get spreadsheet data via Native Mode.
2. Get the result by establishing the EasyBuilder communication setting.

4.1. Acquiring images via Native Mode

When using Native Mode, there are two methods to acquire the result. One is to acquire character strings output from WriteMessage. The other is to acquire the value of spreadsheet.

4.1.1. Using WriteMessage syntax

To receive the character strings output by the WriteMessage, use FormatString function to create the results you want to output, as a character string. A terminator which is selected in the FormatString function window is used when receiving the result of the SendMessageAndWait command. To output, designate character strings created by FormatString function by WriteMessage function.

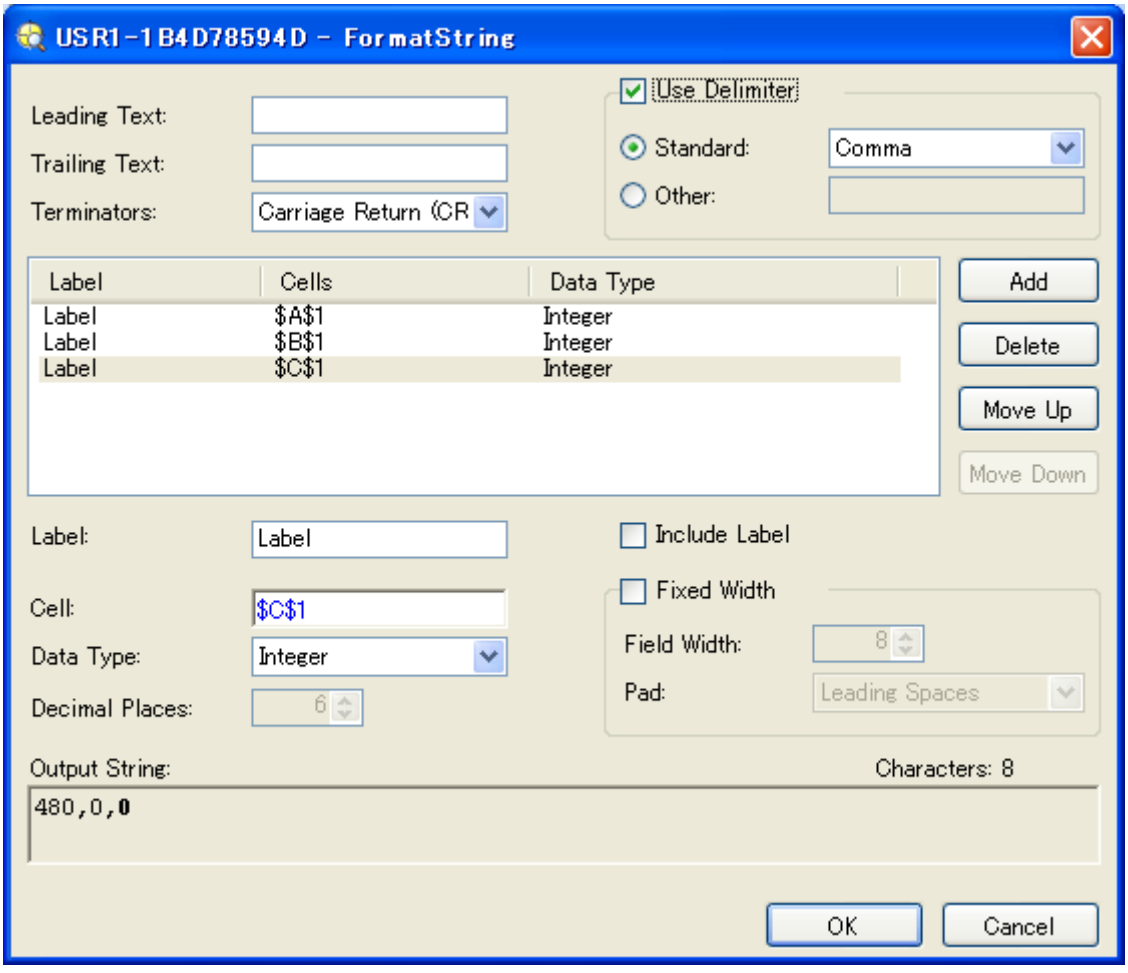

Figure 4-1 FormatString Function

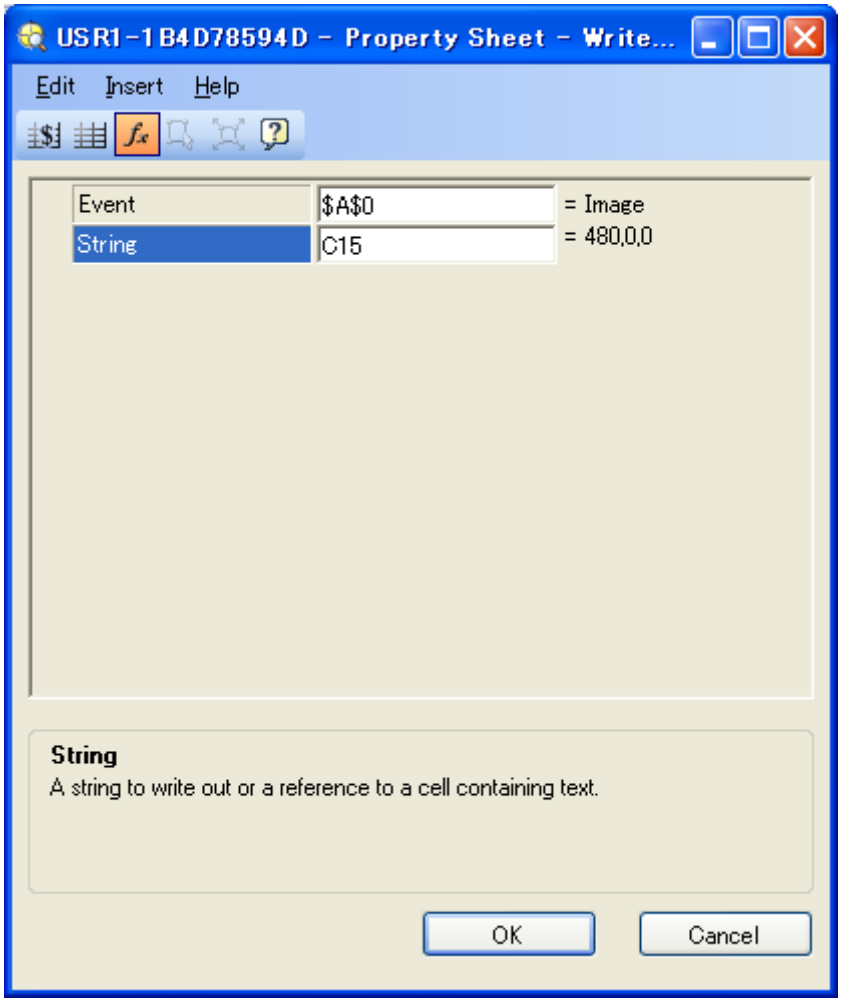

Figure 4-2 WriteMessage Function

Example

```
Private Sub Form_Load()
    Dim caoEng As CaoEngine
    Dim caoCtrl As CaoController

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000")

    Dim strResult As String

    strResult = caoCtrl.Execute("SendMessageAndWait", Array("", 8, 1))
End Sub
```

4.1.2. Acquiring the value of the spreadsheet

In order to acquire the value of spreadsheet, use GetValue command or create variables which refer to the value of cells by AddVariable method.

Example

```
Private Sub Form_Load()
    Dim caoEng As CaoEngine
    Dim caoCtrl As CaoController

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000")

    Dim strResult As String

    Call caoCtrl.Execute("SendMessage", Array("", 8))
    strResult = caoCtrl.Execute("GetValue", Array("0", 60))

End Sub
```

4.2. Acquiring the results by EasyBuilder

In order to acquire the image processing result created by EasyBuilder, use the communication function of the EasyBuilder. By specifying an output result in the communication setting, user can acquire the results by means of SendMessageAndGetCommand, or can receive the result as event output of CaoController. The following shows the way of the communication setting by EasyBuilder.

4.2.1. Setting of the EasyBuilder

Press Communication button shown in Figure 4-3 to open the communication device setting window.

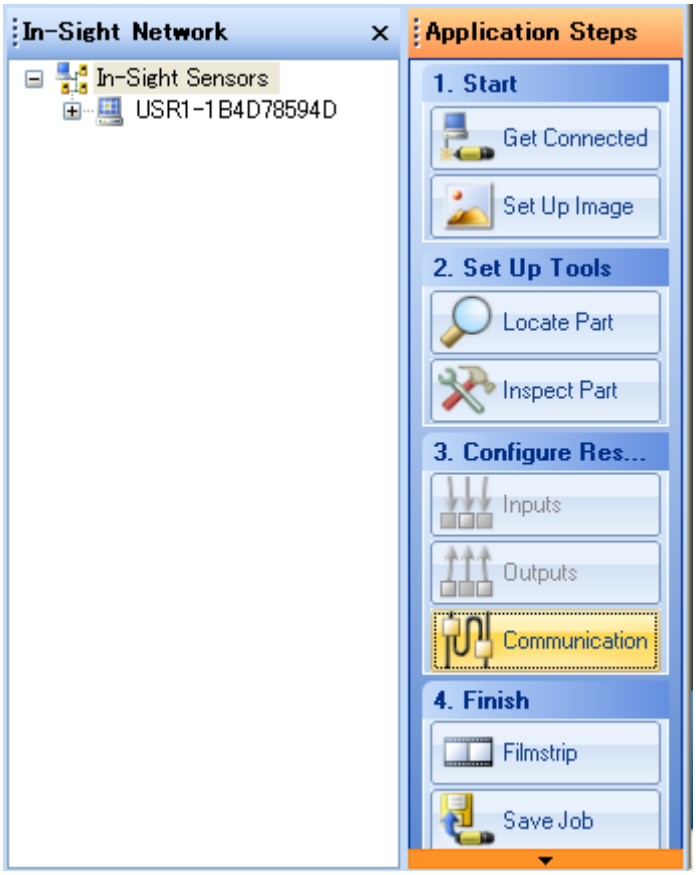

Figure 4-3 Communication Setting

Press Add Device button shown Figure 4-4 to add devices as described below.

Device: Other

Protocol: TCP/IP

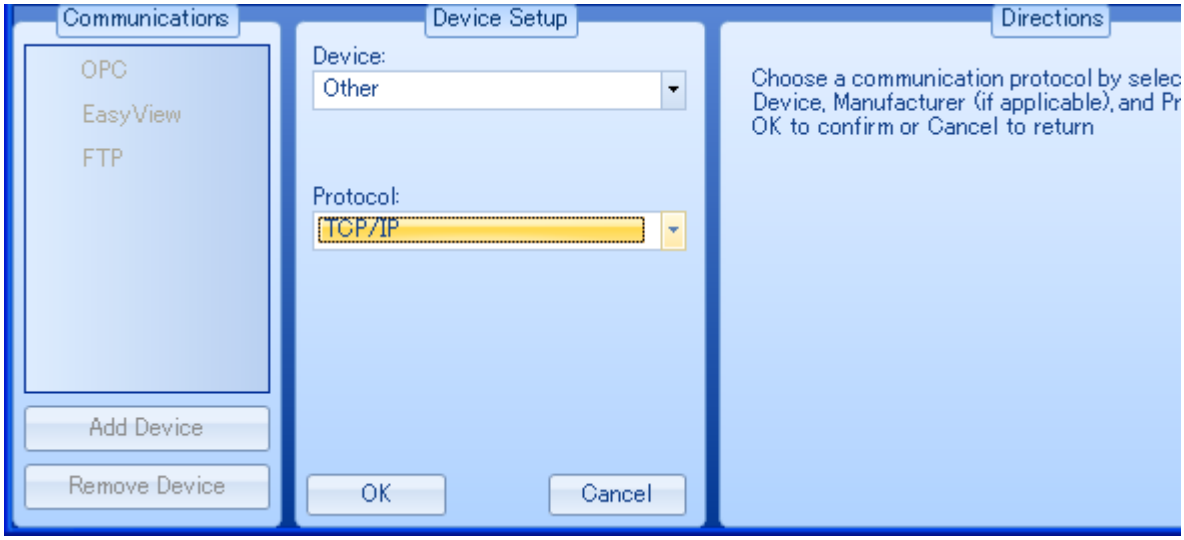

Figure 4-4 Add Device

Specify an Figure 4-5.

Server host name: IP address of the PC which uses provider

Port: Port number specified by EZPort option of AddController

Terminator : CRLF (Fixed)

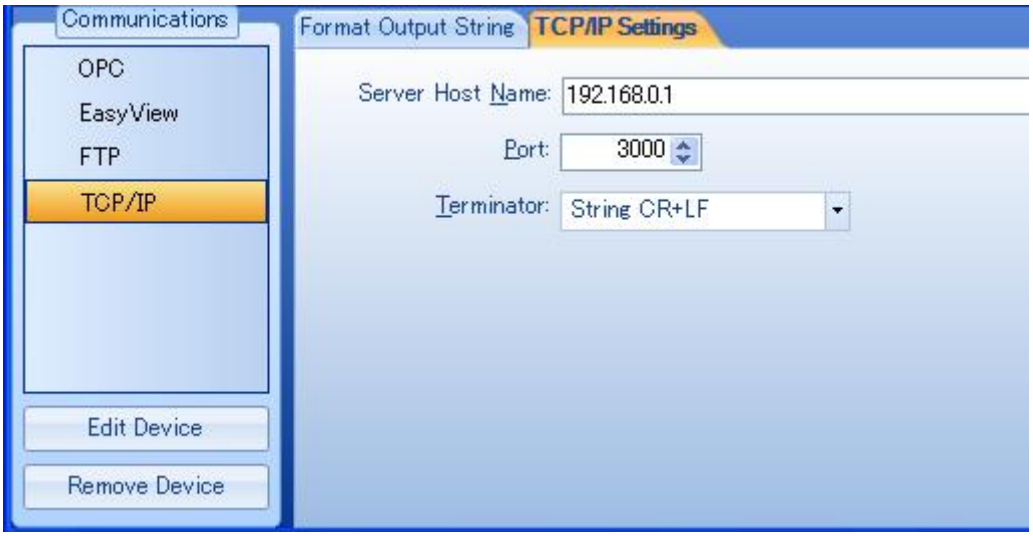

Figure 4-5 TCP/IP Setting

Press the tab of the format output character string, then press the custom format button to put in the result that you want to acquire. Set the terminator as "None".

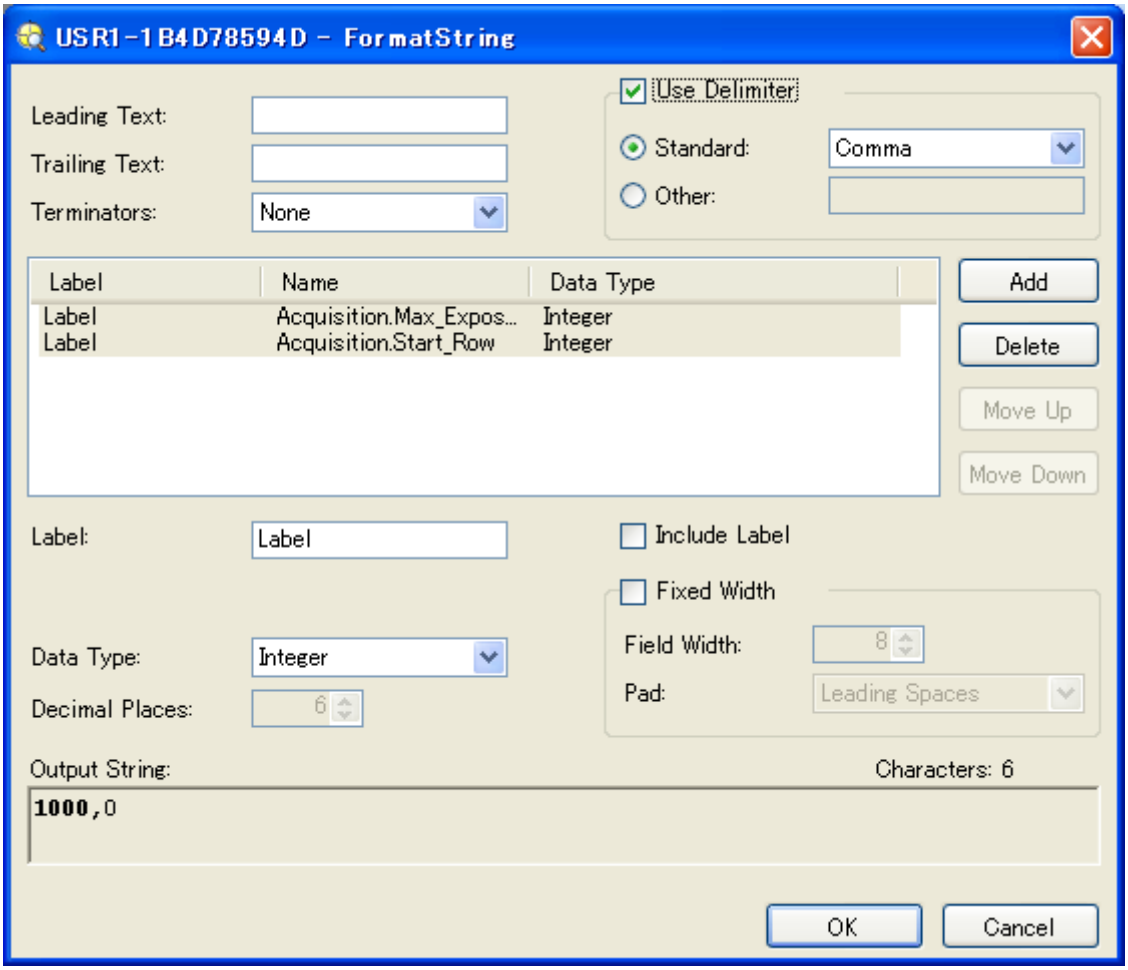

Figure 4-6 Format String

If the online mode is established by above setting procedure, the result will be sent to the preconfigured PC once trigger is activated. The following shows how to receive the result.

4.2.2. How to acquire the result of EasyBuilder

In order to receive the setting result of EasyBuilder, an arbitrary output port number, which is designated in EasyBuilder as "EZPort=Port number", needs to be designated in the option of AddController.

To receive the results, use "SendMessageAndGetEZ" command or, use the combination of "SendMessageEZ" and "GetEZ" command. "SendMessageAndGetEZ" command activates trigger ON. Then, the program will wait until the result is received. "SendMEssageEZ" command activates trigger ON only. Therefore, you need to receive the result by "GetEZ" command.

If the software trigger is not activated, the result is received by OnMessage event of CaoController or using ResetEZ and GetEZ commmands.

There are examples below.

4.2.2.1. Example of the use of SendMessageAndGetEZ

Example

```
Private Sub Form_Load()
    Dim caoEng As CaoEngine
    Dim caoCtrl As CaoController

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")

    Dim strResult As String
    strResult = caoCtrl.Execute("SendMessageAndGetEZ", Array("", 8, 1000))

End Sub
```

4.2.2.2. Example of the use of SendMessageEZ and GetEZ

Example

```
Private Sub Form_Load()
    Dim caoEng As CaoEngine
    Dim caoCtrl As CaoController

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")

    Dim strResult As String
    Call caoCtrl.Execute("SendMessageEZ", Array("", 8))

    ' another transaction

    strResult = caoCtrl.Execute("GetEZ", 1000)
End Sub
```

4.2.2.3. Example of the use of ResetEZ and GetEZ

Example

```
Private Sub Form_Load()
    Dim caoEng As CaoEngine
    Dim caoCtrl As CaoController

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")

    Dim strResult As String
    Call caoCtrl.Execute("ResetEZ")

    'Software trigger
    Call caoCtrl.Execute("SetEvent", 8)


    strResult = caoCtrl.Execute("GetEZ", 1000)
```

```
End Sub
```

4.2.2.4. Example of the use of Event

Example

```
Dim caoEng As CaoEngine
Dim WithEvents caoCtrl  As CaoController

Private Sub Form_Load()
    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("Cog", "CaoProv.Cognex.In-Sight", "",
"conn=eth:127.0.0.1, Timeout=1000, EZPort=3000")
End Sub

Private Sub Command1_Click()
    Call caoCtrl.Execute("SetEvent", 8)
End Sub

Private Sub caoCtrl_OnMessage(ByVal pICaoMess As CAOLib.ICaoMessage)
    Dim strResult As String

    Select Case pICaoMess.Number
    Case 1
        strResult = pICaoMess.Value
        Debug.Print strResult
    End Select
End Sub
```