

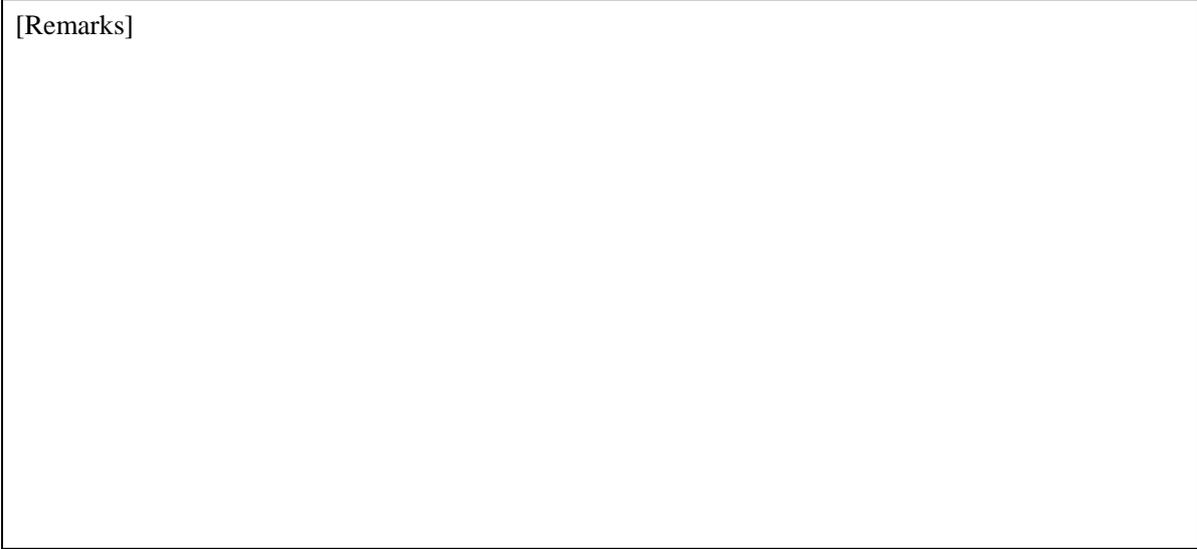
OPC provider Gateway for OPC

Version 1.2.0

User's guide

October 18, 2022

[Remarks]



Contents

1. Introduction.....	4
2. Overview of provider	5
2.1. Overview	5
2.2. Method and Property.....	8
2.2.1. CaoWorkspace::AddController method	8
2.2.1.1. About Connecting to a Remote Server.....	9
2.2.2. CaoController::AddVariable method	10
2.2.2.1. How to use the Name option.....	11
2.2.3. CaoController::get_VariableNames property	12
2.2.4. CaoController::OnMessage event	12
2.2.5. CaoVariable::get_DateTime property.....	14
2.2.6. CaoVariable::get_Value property	14
2.2.7. CaoVariable::put_Value property	14
2.2.8. CaoMessage::Reply method	14
2.3. Variable list.....	15
2.3.1. Controller class.....	15
2.4. Error code	15
3. Sample program.....	16
Appendix A. Settings for OPC server	17
Appendix A.1. Settings for PRO-SERVER manufactured by Digital.....	17
Appendix A.2. Settings for RS-Linx manufactured by Rockwell	18
Appendix A.3. Settings for ISaGRAF manufactured by ICS Triplex	20
Appendix A.4. Settings for DeviceXPlorer manufactured by TAKEBISHI	21

1. Introduction

This is a user's guide for "CAO provider for OPC connection (hereafter OPC provider)", which establishes a connection between CAO with PLC (Programmable Logic Controller) through OPC (OLE for Process Control) server.

With this provider, CAO-compatible tools can access PLC and/or display that equip OPC server, as well as robots.

Because the purpose of OPC client, which works as a CAO provider, is to cover the CAO interface, it does not support OPC original functions. We will study the necessity of OPC original functions by every revision of CAO interface.

2. Overview of provider

2.1. Overview

OPC provider is a CAO provider to connect to PLC via CAO. (Figure 2-1). With this provider, CAO-supporting tools can access not only robots but also PLCs as well.

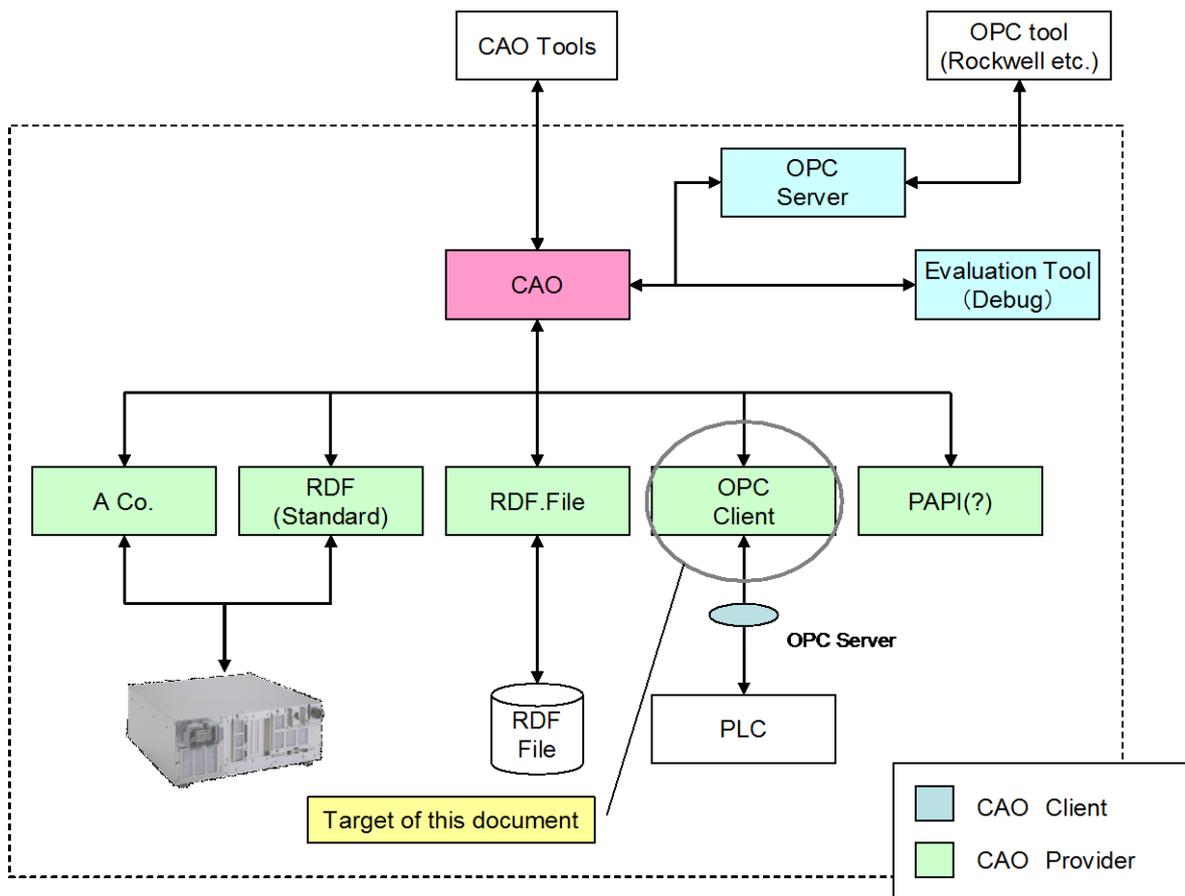


Figure 2-1 CAO provider for OPC connection

This OPC provider allows CaoVariable objects to refer to values of Item in OPC server (Figure2-2). However, only the data reading/writing of synchronous type are supported because the interface and object structure are different between OPC server and CAO. The data reading/writing of asynchronous type are not supported by this provider because it has already achieved by the function of CAO engine.

Moreover, the alarm state of the Item value can be observed as a function of AE, and the event be received.

- It corresponds to DA(DataAccess) of OPC and AE(Alarm&Event) in this provider.

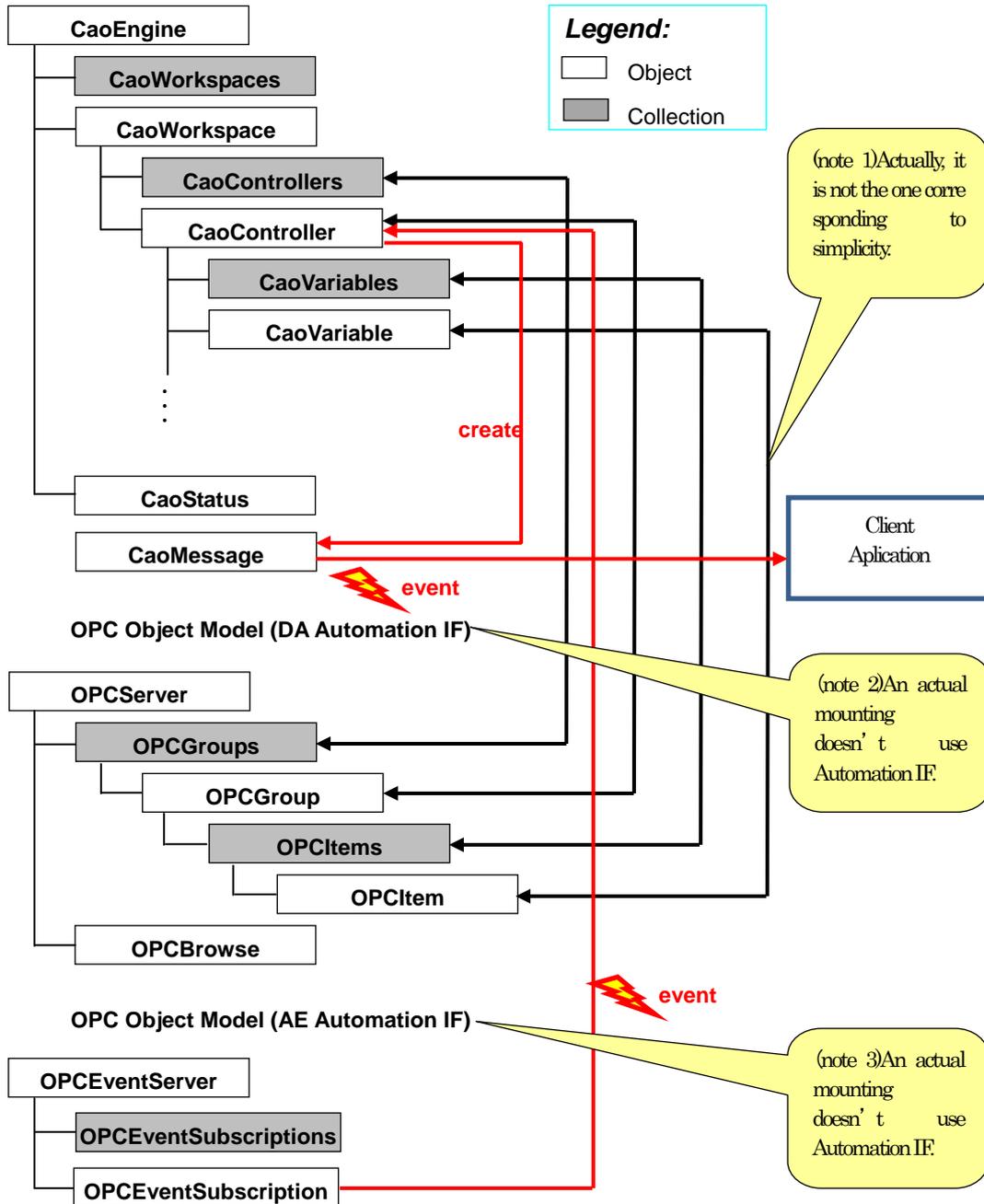
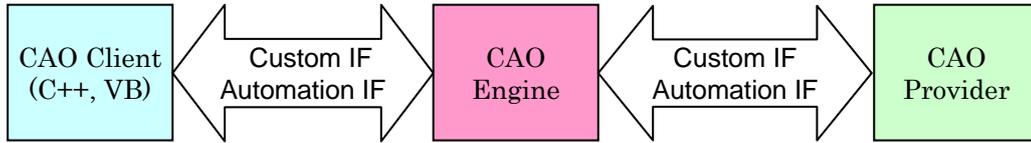


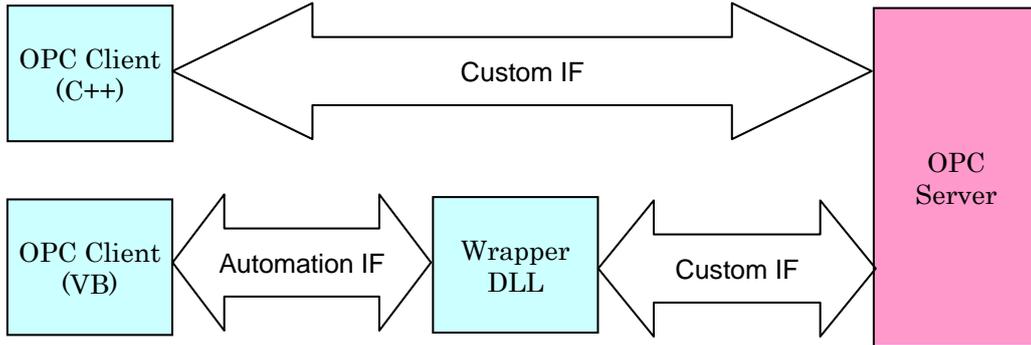
Figure2-2 Object mapping

OPC provider uses DA custom interface whereas it does not use DA automation interface. Please note that, in OPC specification, the object model of these two interfaces are different (Figure2-3).

□ CAO Module



□ OPC Module



■ Problem in OPC composition

1. When custom IF and automation IF are used, an operation different in the language that develops the client application program is needed because even the object model is different.
2. When even a similar function is not individually mounted, the vender who offers the server doesn't become it because it has not divided into engine and provider part CAO.

Figure2-3 Comparison of module structure of CAO and OPC

Table 2-1 OPC provider

File name	CaoProvOPC.dll
ProgID	CaoProv.OPC
Registration ¹	regsvr32 CaoProvOPC.dll
Deregistration	regsvr32 /u CaoProvOPC.dll

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

2.2. Method and Property

2.2.1. CaoWorkspace::AddController method

The following shows the specification of AddController method of CaoWorkspace class of CAO.

```

AddController
(
  "< Controller name >",           // Controller name
  "CaoProv. OPC",                 // Provider name. Fixed.
  "< computer name >",           // Computer name where provider executes.
  "< option >"                   // Option character string(OPC option)
)
    
```

For provider name, use "CaoProv.OPC" whenever you use this provider. For <computer name >, enter the computer name where this provider executes.

The format of < option > is as follows.

```

<Option> ::= OPCServer=<OPC server name> [, DeadBand =<setting value> [,
UpdateRate=<setting value>] [, AccessPath=<setting value >] [, ActiveState=<True|False>]
[,TimeBias=<setting value >][,LocaleID=<setting value>][, Machine=<computer name where OPC
server executes>]]
    
```

An OPC group is created with the name of "CaoProv_<controller name>".

You must specify <OPC server> always. Enter the name of OPC server (ProgID) for this entry. Parameters enclosed by "[]" are not prerequisite items, and default values will be used if not specified. The specified value is interpreted based on the OPC server being called. For other options, refer to Table2-2.

Table2-2 Option character string of CaoWorkspace::AddController

OPC option name	Description
OPCServer=< OPC server >	ProgID of OPC server . [Mandatory]
DeadBand[=< DeadBand >]	Deadband (%). This option is used when a Group object of OPC is created (default: 0.0).
UpdateRate [=< Update period >]	Update interval (ms) of Group. This option is used when a Group object of OPC is created. (default: 1000)
AccessPath [=< AccessPath >]	Access path for initial setting. You can specify Access path from AccessPath option of AddVariable individually. The value of this entry is effective when you do not specify AccessPath in AddVariable.(default: Null string)

ActiveState[=<True False>]	Setting of ActiveState. True: Set the added item to Active. False: Set the added item to Inactive. You can set the ActiveState from ActiveState option of AddVariable individually. The value of this entry is effective when you do not specify ActiveState in AddVariable. (default: True)
TimeBias[=<Difference adjustment time >]	Adjust the time difference between devices and OPC (min). (default: 0)
LocaleID [=< locale ID >]	Locale ID setting. (default: Locale ID set with CaoConfig)
Machine [=< computer name where OPC server executes>]	OPC server-execution-machine-name (IP addressable). Starts up as a local server if the specified machine name is the machine that runs the provider, or as a remote server if it is a different machine.(Default : "localhost") Refer to 2.2.1.1 for the connection to the remote server.
EventAutoACK[=<True False>]	The operation when the condition event of OPC is received is specified. (default: True) True: The ACK response is automatically returned. False: The ACK response is not automatically returned. (Respond by the Reply method of OnMessage.) - Only at connection to OPC AE server

The following sample program shows how to execute AddController method.

```

AddController
(
    "OPC1",
    "CaoProv. OPC", // Provider name.
    "" // Start as In-process.
    "OPCServer= OPC.DAServer, UpdateRate=1000" // Update interval is one second.
);
    
```

2.2.1.1. About Connecting to a Remote Server

To connect to a remote server, one of the following conditions must be met:

1. OPCEnum service is registered on OPC server². (Preferred)
2. The class ID of OPC servers is registered PC the providers are run

Otherwise, the connection fails.

² The architecture of OPC servers (32/64bit) must be supported.

2.2.2. CaoController::AddVariable method

The following shows the specification of CaoController::AddVariable method of CAO.

```
CaoController::AddVariable
(
  "< variable name >"           // Variable name
  "< option >"                 // Option string
)
```

Set this argument as follows when you use the OPC provider.

< variable name >::=< item ID >

<option> ::= [[AccessPath=<Access path>] [, ActiveState=<True|False>] [,RequestType=<variable type to receive>] [,Source=<data source>]]

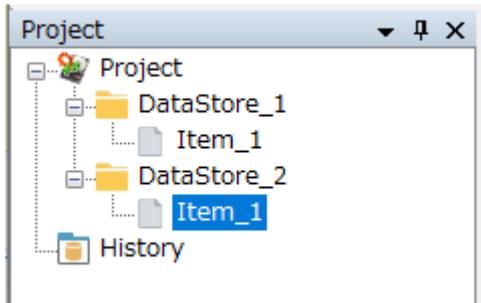
Parameters enclosed by “[]” are not prerequisite items, and default values will be used if not specified. The specified value is interpreted based on the OPC server being called. For variable type to receive, use VT_TYPE. For details about VT_TYPE and supported data types are described in Table2-4. For other options, refer to Table2-3.

Table2-3 Option character string of CaoController::AddVariable

OPC option name	Description
Name [=<Tag name>]	Specify the target tag name. If this option is omitted, the variable name will be used as the tag name. See 2.2.2.1 for how to use it.
AccessPath [=< Access Path>]	Access path. (default: AccessPath option value of parents controller) Overwrite the value configured by AccessPath option of AddController.
ActiveState [=<True False>]	Setting of ActiveState. (default: ActiveState option value of parents controller) Overwrite the value configured by ActiveState option of AddController. True: Set the added item to Active. False: Set the added item to Inactive.
RequestType [=<variable type to receive >]	Setting of variable type to receive data (default: VT_EMPTY)

<p>Source [=<Data Source >]</p>	<p>Setting of data source. (default: When ActiveState is set to “active”, this will be OPC_DS_CACHE. When ActiveState is set to “inactive”, this will be OPC_DS_DEVICE.)</p> <p>1:OPC_DS_CACHE 2:OPC_DS_DEVICE</p>
---------------------------------------	--

2.2.2.1. How to use the Name option



If you have a CaoOPC server like the one above, you can use one controller

- Item_1 of DataStore_1
- Item_1 of DataStore_2

To use at the same time, use the Name option and add variables as shown below.

```

Dim eng As CaoEngine
Dim ws As CaoWorkspace
Dim ctrl As CaoController
Dim var1 As CaoVariable
Dim var2 As CaoVariable

Set eng = new CaoEngine
Set ws = eng.Workspaces(0)
Set ctrl = ws.AddController("", "CaoProv.OPC", "", "OPCServer=OPC.CAO")
Set var1 = ctrl.AddVariable("ITEM-1", "AccessPath=DataStore_1, Name=Item_1")
Set var2 = ctrl.AddVariable("ITEM-2", "AccessPath=DataStore_2, Name=Item_1")
. . .
    
```

Table2-4: VARIANT type data (excerpt)

Data type	Code	Size (byte)	Description
VT_I2	2	2	Single precision (16-bit) integer

VT_I4	3	4	Double precision (32-bit) integer
VT_R4	4	4	Single precision (32-bit) floating point decimal
VT_R8	5	8	Double precision (64-bit) floating point decimal
VT_CY	6	8	Currency type. Same as VT_UI8.
VT_DATE	7	8	The number of days that have passed since 1899/12/30. Same as VT_R8.
VT_BSTR	8	(variable)	Character string Add UNICODE characters and null terminator after the characters written in VT_UI4.
VT_BOOL	11	1	Same as VT_UI1. 0: FALSE, Other than 0: TRUE
VT_VARIANT	12	(variable)	Variant type (Use only for the Variant type array.)
VT_UI1	17	1	Character unsigned
VT_ARRAY	8192	(variable)	One dimensional array consists of above-mentioned data type.

2.2.3. CaoController::get_VariableNames property

This property acquires the system variable list.

2.2.4. CaoController::OnMessage event

The following OnMessage events are generated in this provider.a

- Only at connection to OPC AE server

Number	Description
1	Simple event
2	Tracking event
3	Condition event

• Simple event		
Occurrence condition	When you fill the occurrence condition of the simple event set with the server by the change of the value.	
Number	1	
Value	Index : 0 Type : VT_DATE	Date
	Index : 1 Type : VT_I4	Severity (1~1000)

	Index : 2 Type : VT_BSTR	Message
Description	Date : Date when event was generated Severity : Severity of event set with server Message : Message of event set with server	

• Tracking Event		
Occurrence condition	When you fill the occurrence condition of Tracking Event set with the server by the change of the value.	
Number	2	
Value	Index : 0 Type : VT_DATE	Date
	Index : 1 Type : VT_I4	Severity (1~1000)
	Index : 2 Type : VT_BSTR	Message
	Index : 3 Type : VT_BSTR	Source
Description	Date : Date when event was generated Severity : Severity of event set with server Message : Message of event set with server Source : Variable where event is generated	

• Condition Event		
Occurrence condition	When you fill the occurrence condition of the condition event set with the server by the change of the value.	
Number	3	
Value	Index : 0 Type : VT_DATE	Date
	Index : 1 Type : VT_I4	Severity (1~1000)
	Index : 2 Type : VT_BSTR	Message
	Index : 3 Type : VT_BSTR	Source

	Index : 4 Type : VT_BSTR	Condition Name
	Index : 5 Type : VT_BSTR	Subconditon Name
	Index : 6 Type : VT_I4	Condition ID
	Index : 7 Type : VT_BOOL	Flag
Description	Date : Date when event was generated Severity : Severity of event set with server Message : Message of event set with server Sourece : Variable where event is generated Condition Name : Condition name of generated condition event Subconditon Name : Subcondition name of generated condition event Condition ID : Value in which generated condition event is identified Flag : It is shown that the object condition was deleted for FALSE.	

2.2.5. CaoVariable::get_DateTime property

This get_DateTime property acquires the timestamp of Item ID.

Note that the timestamp acquired by this property is the one at the execution of get_Value property.

2.2.6. CaoVariable::get_Value property

This get_Value property acquires the value of item ID.

2.2.7. CaoVariable::put_Value property

This put_Value property sets the value of item ID.

2.2.8. CaoMessage::Reply method

The ACK response can be returned to the generated alarm by using the Reply method of the CaoMessage class.

- Only at connection to OPC AE server

2.3. Variable list

2.3.1. Controller class

Table 2-5 Controller class System variable list

Variable name	Data type	Description	Attribute	
			get	put
@SERVER_STATE	VT_I4	Get the current state of OPC server. 1 : OPC_STATUS_RUNNING 2 : OPC_STATUS_FAILED 3 : OPC_STATUS_NOCONFIG 4 : OPC_STATUS_SUSPENDED 5 : OPC_STATUS_TEST	✓	-

2.4. Error code

OPC provider defines the following original error code. For about ORiN2 common errors, refer to the error code section of "[ORiN2 Programming guide](#)".

Table 2-6 Original error code list

Error name	Error number	Description
E_OPC_QUALITY_NOT_GOOD	0x80100800	The state of Item is not "GOOD".
E_OPCAE_ACK_NOTCOMPATIBLE_EVENT	0x80100801	It is not possible to respond.
E_OPCAE_ACK_EVENTAUTOACK_ENABLED	0x80100802	An automatic response is set.

3. Sample program

This chapter shows a sample program to connect to CaoOPC, set the character string of textbox to Item ID:item1 with a Push button, and then get the value of Item ID:item1 with a Get button.

List 3-1 Sample.frm

```
Option Explicit
Dim eng As CaoEngine
Dim ws As CaoWorkspace
Dim ctrl As CaoController
Dim item As CaoVariable

Private Sub Form_Load()
    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    ' Obtain a variable object.
    Set ctrl = ws.AddController("", _
        "CaoProv.OPC", _
        "OPCServer=OPC.CAO, UpdateRate=500")
    Set item = ctrl.AddVariable("item1", _
        "AccessPath=OPCC2")

End Sub

Private Sub cmdGet_Click()
    ' Obtain a value.
    txtValue.Text = item.Value

End Sub

Private Sub cmdPut_Click()
    ' Obtain a value.
    item.Value = txtValue.Text
    txtValue.Text = ""

End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set item = Nothing
    Set ctrl = Nothing
    Set ws = Nothing
    Set eng = Nothing

End Sub
```

Appendix A. Settings for OPC server

Appendix A.1. Settings for PRO-SERVER manufactured by Digital

OPC server manufactured by Digital has two types; In-process type and Local/Remote process type. This example explains In-process type. Since CAO itself already supports both local process and remote process, users of this provider need In-process type only. In this case, two processes (CAO process and PRO-SERVER process) are generated.

Specify arguments of AddController method and AddVariable method as follows.

OPCServer option of AddController ::= "DIGITAL.OPCPRO1"

Variable name of AddVariable ::= "<port name >:< address >"

For instance, if you enter "GP1:LS 0022:2", it means that the word data of address LS0022 of GP1 is specified. For the address, you can directly specify the address of PLC that is connected to the display as well. For example, when "SLC5" manufactured by Allen-Bradley is connected, if you specify "GP1:N 027007:2", it means that N027007, which is the address of SLC5, is specified.

For detailed specification of OPC server, refer to Help (opcpro.hlp).

- Content of OPCPRO.ini file

[Server]

Server_Max=10 <- (note) The number of instance is limited.

Group_Max=10

Item_Max=100

InproSvr_ScanRate=250

LocalSvr_ScanRate=500

[Client]

CharacterStringCode=0

ItemID_Separator=. :!

Appendix A.2. Settings for RS-Linx manufactured by Rockwell

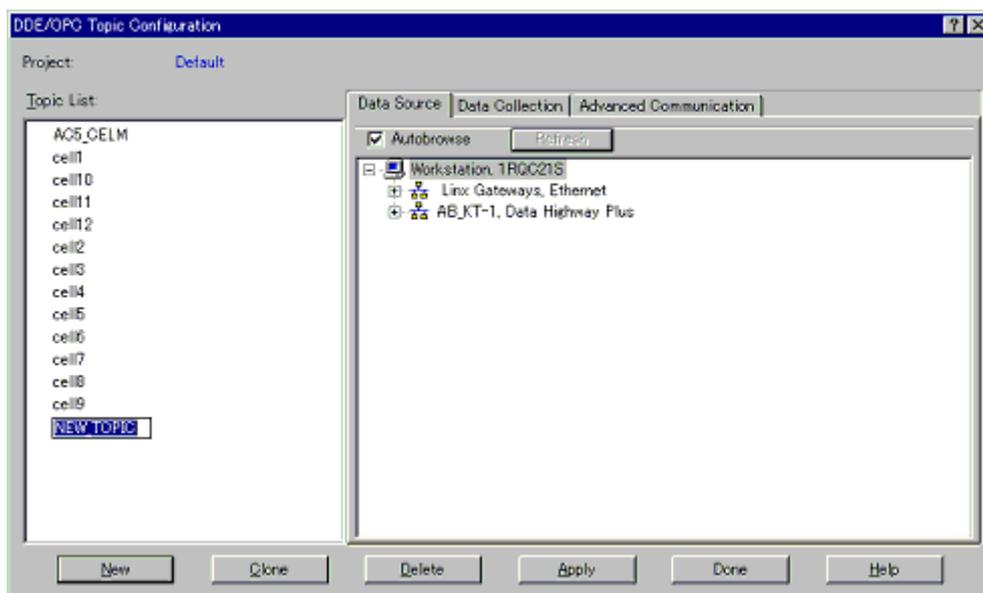
Before using RS-Linx manufactured by Rockwell as an OPC server, you need to prepare some settings with RS-Linx (see below). Once all preparations complete, specify arguments for AddController method and AddVariable method as follows.

OPC server option of AddController ::= "RSLinx OPC Server"³

Variable name of AddVariable ::= "< topic name >! < address >"

For instance, when the default topic name is assumed to be topic1, specify the option character string of AddController method as "OPCServer=RSLinx OPC Server, AccessPath=topic1". Also, if you specify the variable name of AddVariable method as "topic1!N027:041" and the option string as "RequestType=2", the word data of address "N027:041" of topic1 is specified. (Because the default topic name is specified, you can omit "topic1" of the variable name.)

- The following shows the setup process of RS-Linx.⁴
 1. Create a new topic with [Topic Configuration...]⁵. Use any character strings for the topic name.
 2. Select the topic you've just created. From the [Data Collection] tab, specify the processor type. If the processor is SLC5/04, select [SLC503+]
 3. Specify Communication Driver in the Advanced Communication tab.
 4. Specify the Node address of connection destination PLC for Station in the Advanced Communication tab.



³ Because ProgID of RSLinx includes blanks, enclose them with round brackets. OPCServer=(RSLinx OPC Server)

⁴ Some packages of RS-Linx do not support OPC functions. Depending on the version, setting process may differ.

⁵ This menu is enabled only when DDE project opens

Figure3-1: Topic Configuration dialog

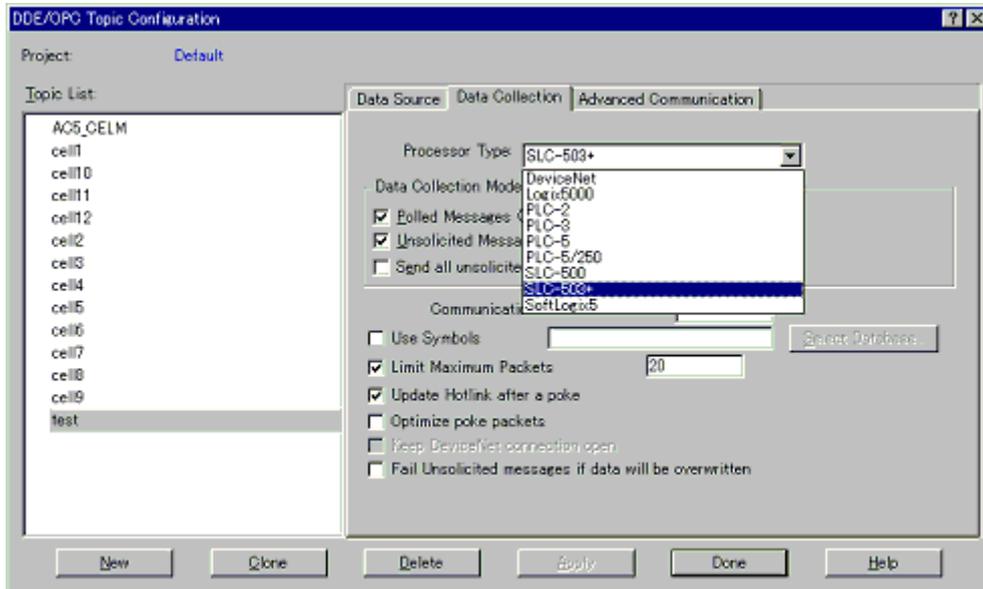


Figure3-2: Data Collection tab

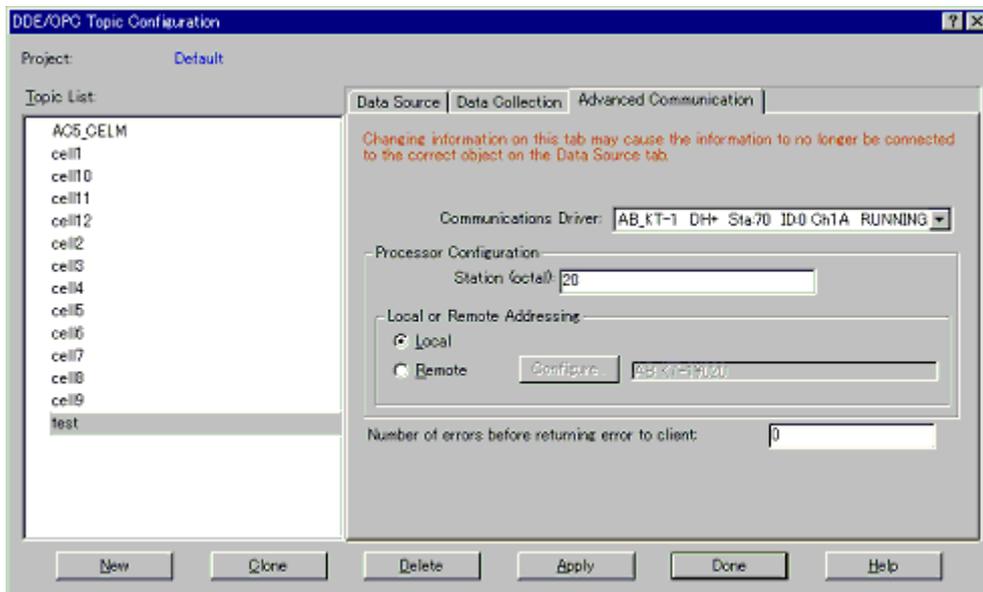


Figure3-3: Advanced Communication tab

Appendix A.3. Settings for ISaGRAF manufactured by ICS Triplex

Before using ISaGRAF manufactured by ICS Triplex as an OPC server, you need to prepare some settings with ISaGRAF (see below). Once all preparations complete, specify arguments for AddController method and AddVariable method as follows.

OPC server option of AddController ::= "ICSTriplex.IsagrafOPCDA20.1"

Variable name of AddVariable ::= "<configuration name>.<resource name>.<variable name>"

For instance, if you enter "Config1.resource1.var1", it means that the variable var1 of resource1 in config1 is specified.

- The following shows the setup process of ISaGRAF.
 1. Start Workbench.exe of ISaGRAF.
 2. From [Open], open the project file.
 3. From the Menu, select [Debug] – [Simulation].

Appendix A.4. Settings for DeviceXPlorer manufactured by TAKEBISHI

Before using DeviceXPlorer manufactured by TAKEBISHI as an OPC server, you need to prepare some settings with DeviceXPlorer. Once all preparations complete, specify arguments for AddController method and AddVariable method as follows.

OPC server option of AddController ::= "Takebishi.DXP "

Variable name of AddVariable ::= "< path>.<tag name >"

For instance, if you enter "Device1.LoopFlag", it means that LoopFlag in Device1 is specified.

