

CaoOPC (OPC Server)

User's guide

Version 1.5.2

March 12, 2021

[Remarks]

This application complies with OPC DA (Data Access) 2.0. Note that OPC DA1.0 specification is not supported although it has been implemented

Contents

1. Introduction.....	4
2. Overview of CaoOPC	5
2.1. Overview	5
2.2. OPC interface supporting status	7
2.3. Data type supporting status	8
2.4. Item ID and AccessPath.....	8
2.5. Operation model	8
2.6. LocaleID.....	9
2.7. Property ID list	10
2.8. Sample program	10
3. CaoOPCConfig.....	13
3.1. Overview	13
3.2. Operation method	14
3.2.1. Tab input.....	14
3.2.1.1. Engine tab.....	14
3.2.1.2. Controller tab	16
3.2.1.3. Item tab.....	17
3.2.2. Menu	21
3.2.2.1. File menu	21
3.2.2.2. Edit menu.....	22
3.2.2.3. Action menu	23
3.2.2.4. Help menu.....	25
3.3. Registry structure information	25
4. OPC provider.....	26
4.1. Overview	26
4.2. Method property.....	27
4.2.1. CaoWorkspace::AddController method	27
4.2.2. CaoController::AddVariable method	27

1. Introduction

This book is a user's guide for CaoOPC that performs as a client application of CAO (Controller Access Object).

CaoOPC is an OPC server that regards CAO module as an FA equipment, and provides various OPC (OLE for Process Control) functions (Figure 1-1). Because CAO is a module that enables communication between various FA devices, CaoOPC that is an application on the upper layer of CAO can consequently connect with a variety of FA devices.

In addition, because CaoOPC provides OPC server function, it can use commercial OPC client software. It means, implementing CAO provider enables users to use OPC client application as well as CAO client application. This enables wide variety of system development.

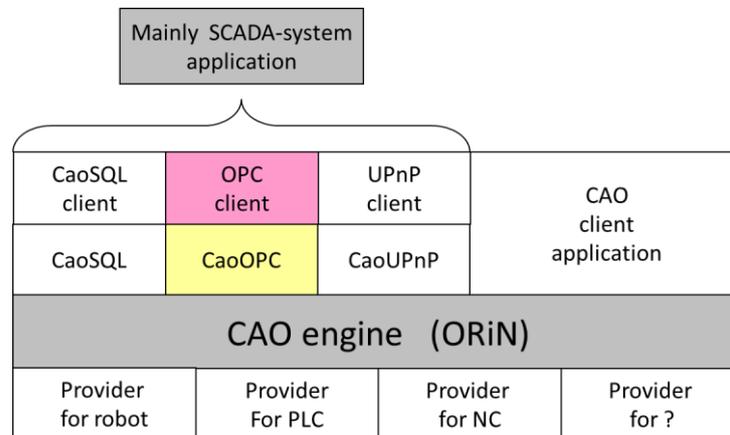


Figure 1-1 Form of CAO application

OPC is mainly considered as “a server software that provides process data to clients”. This idea is different from that of CAO which models all resources that FA devices have. Therefore, CaoOPC is mainly implemented only with CAO's variable class. As a result, CaoOPC-used application is naturally limited to SCADA application.

On the other hand, using CaoSQL and/or CaoOPC makes it easier to develop SCADA application than using CAO directly. CaoSQL offers advanced functions, such as data base access function that is indispensable to SCADA application, so that CaoOPC can use commercial OPC client software.

This document mainly describes how to use CaoOPC. For details about OPC specification, refer to [OPC Foundation](#) website. For the way of how to use OPC server from CAO client application, refer to "[OPC provider user's guide](#)".

2. Overview of CaoOPC

2.1. Overview

The figure below shows the structure of CaoOPC. CaoOPC is a middleware that connects OPC client with CAO. CaoOPC works as an OPC server that processes requests from OPC client by using CAO functions. CaoOPC is an independent process (out process server) and it provides service to several OPC clients.

Since CaoOPC ver.1.0 supports OPC DA(Data Access) 2.0, users can use OPC clients that use functions of DA2.0 as well. ProgID of CaoOPC is "OPC.CAO".

CaoOPCConfig is a configuration tool that configures motion option of CaoOPC and parameters necessary to create CaoController object and CaoVariable object. The contents configured by this tool will be stored in registry and csq file and will be referred at the start of CaoOPC.

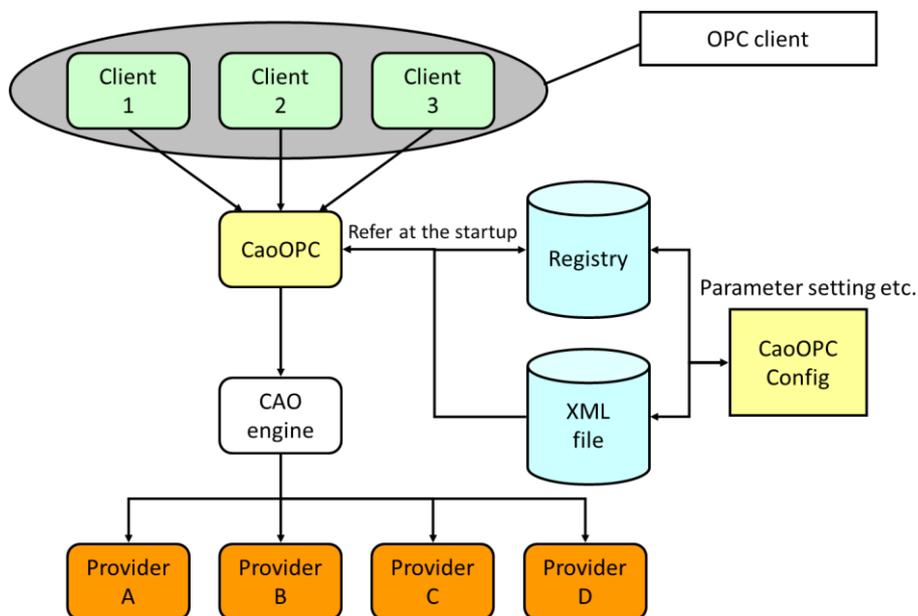


Figure 2-1 CaoOPC diagram

For example, if you use a CAO provider designed for OPC, CaoOPC is inserted between OPC client and OPC server as Figure 2-2 shows. As explained earlier, the advantage for OPC client of using CaoOPC is the accessibility to various FA devices (such as robots) that are offered by CAO providers.

In addition, even you just connect OPC client and OPC server via CAO provider, there is another advantage. CAO provider enables OPC client to connect to OPC server via the Internet. This connection mechanism is an application of ORiN's CAP (Controller Access Protocol) technology

As described above, by simply inserting CaoOPC between OPC client and OPC server, still user can enjoy the advantage of ORiN technology. (Also, using CRD provider enable OPC client to access CRD (XML) type files.)

☆ Inserting ORiN between OPC client and server

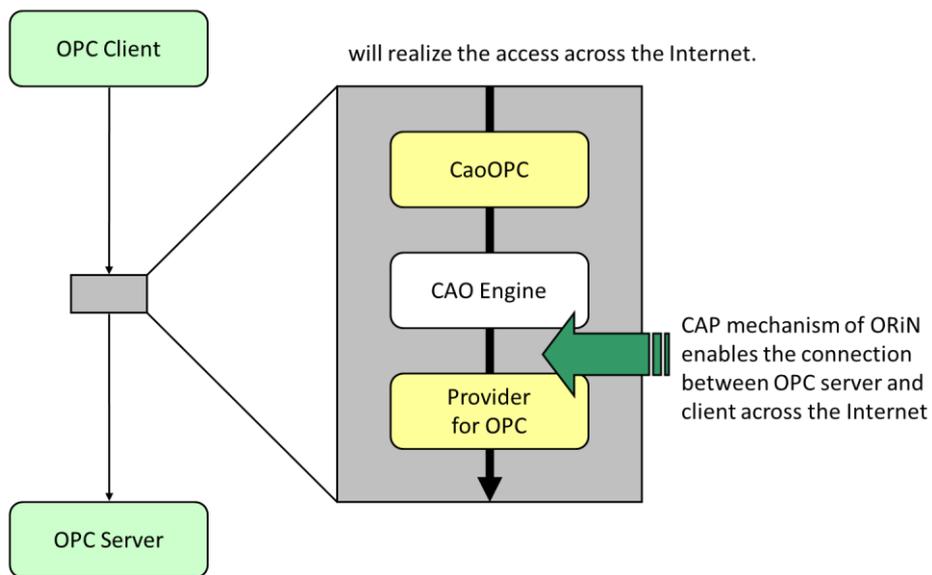


Figure 2-2 Insert CaoOPC between OPC client and OPC server

2.2. OPC interface supporting status

Table 2-1 shows the OPC's supporting status for DA2.0 custom interface.

Table 2-1 OPC interface supporting status

OPC interface	Supported	Remarks
+ OPC server object		
IOPCCommon	Yes	
IOPCServer	Yes	
IOPCServerPublicGroups	No	Option
IOPCBrowseServerAddressSpace	Partly supported	Option ¹
IOPCItemProperties	Yes	² ³
IConnectionPointContainer	Yes	
IPersistFile	No	Option
+ OPC group object		
IOPCGroupStateMgt	Yes	
IOPCPublicGroupStateMgt	No	Option
IOPCSyncIO	Yes	
IOPCAsyncIO	Yes	Old Interface
IOPCItemMgt	Yes	
IDataObject	Yes	Old Interface
IOPCAsyncIO2	Yes	
IConnectionPointContainer	Yes	
+ OPC item enumeration object		
IEnumOPCItemAttributes	Yes	

¹ BrowseAccessPath method is not supported. For character string filter of BrowsOPCItemIDs method, only "*" is available.

² Only items on the default controller and/or that of expressed by full path are available.

³ For about available property IDs, refer to "2.7 Property ID list".

2.3. Data type supporting status

CaoOPC retrieves values from CAO as VARIANT type. It means that the data type of CaoOPC depends on the implementation of the provider, including arrays. However, because the value change is checked by VarComp function, it is impossible to detect the change of values on arrays.

In OPC specification, client can specify the data type to receive. To follow this specification, CaoOPC needs to convert data type inside. Because VariantChangeType function (Automation API) is used for this processing, it is also impossible to convert arrays as well.

2.4. Item ID and AccessPath

CaoOPC supports AccessPath. For AccessPath, *Name* of Controller (CaoOPC controller name)⁴ configured by CaoOPCConfig tool is used. You can specify AccessPath by using IOPCItemMgt::AddItem function or by the following entry.

item ID = [<AccessPath>!]<ItemID>

For instance, if AccessPath is "PLC1", and ItemID is "W100", this is expressed as "PLC1!W100". In this case, CaoOPC uses "W100" as a variable name of CaoVariable. However, it can be the case that the delimiter "!" is already included in the variable name of CaoVariable, therefore, we recommend to use AddItem function to specify AccessPath.⁵

Item ID can be dynamically added as well. In this case, the native type of the item should be VT_VARIANT and the class should be Controller class.⁶

2.5. Operation model

The operation model of CaoOPC is shown in Figure 2-3. Inside of the processing, values of "Item", which are registered in the Item list, are read from CAO all the time in the interval specified by UpdateRate and cached. Then, a retrieved value is compared to the cache, and if these are unmatched, the cache is overwritten and onDataChange event is generated⁷.

In general, values are read by OPC_DS_CACHE option. In this case, cached values are retrieved. If values are read by OPC_DS_DEVICE option, regardless of the UpdateRate interval, values are retrieved from CAO at the timing of Read function calling. For both cases, Item must be "Active" in order to update values.

On the other hand, values are written into CAO at the timing of a client calling Write function.

⁴ Please note that this is not a ControllerName (Cao controller name)..

⁵ To specify Item ID including delimiter "!", you must enter the controller name. (Default controller name cannot be used.)

⁶ When CaoOPC closes, items dynamically added will be released.

⁷ This uses VarComp to judge the change of values. Therefore, if any data type that VarComp cannot recognize (VT_I1, VT_UI2, and VT_UI4, etc.) is used, onDataChange event is not generated.

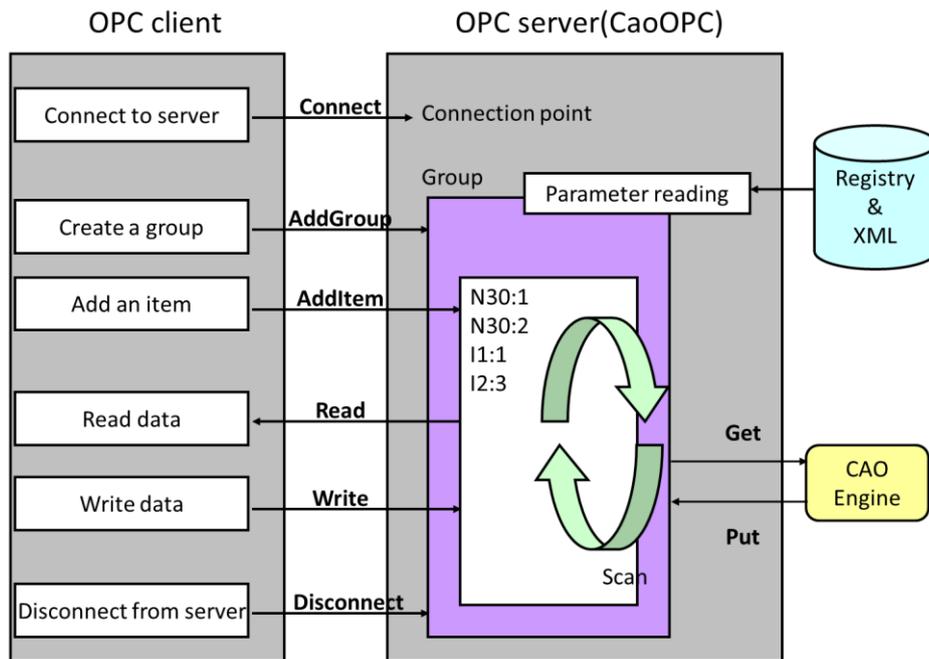


Figure 2-3 CaoOPC working model

2.6. LocaleID

LocaleID is an integer of the DWORD type that represents a language. See Table 2-2.

Table 2-2 LocaleID

Locale name	Language code	Hexadecimal	Decimal
Japanese	ja	0x0411	1041
English (U.S.)	en-us	0x0409	1033
German (Germany)	de	0x0407	1031

You can set the default value of CaoOPC LocaleID by either using the CaoOPCConfig tool or IOPCCommon::SetLocalID function.

IOPCCommon::QueryAvailableLocalIDs function returns a list of three elements at any time; LOCALE_SYSTEM_DEFAULT (system default locale ID:2), LOCALE_USER_DEFAULT (user's default locale ID:1), and LOCALE_NEUTRAL(0).

When a Group object is created from the OPC server, a value specified by the argument of the IOPCServer::AddGroup function will be the default value of the group. You can change this value by IOPCGroupStateMgt::SetState function. A group's LocaleID is used to detect the value change of Item that is a member of this group⁸

⁸ Precisely to say, it is used in the third argument of the VarCmp function of Automation library (oleaut32.lib) of Platform SDK.

2.7. Property ID list

The following tables show the property ID that IOPCItemProperties method can retrieve. For details about each property, refer to the specifications of OPC.

Table 2-3 Property ID list (Part 1)

ID	Data type	
1	VT_I2	"Item Canonical DataType"
2	<varies>	"Item Value"
3	VT_I2	"Item Quality"
4	VT_DATE	"Item Timestamp"
5	VT_I4	"Item Access Rights"
6	VT_R4	"Server Scan Rate"

Table 2-4 Property ID list (Part 1)

ID	Data type	
100	VT_BSTR	"EU Units" e.g. "DEGC" or "GALLONS"
101	VT_BSTR	"Item Description"
102	VT_R8	"High EU"
103	VT_R8	"Low EU"

2.8. Sample program

This section shows sample programs of CaoOPC. The programming language used is Microsoft Visual Basic6.0. Please check that OPC Automation wrapper DLL has been registered. Otherwise, the program does not run.

(1) Addition of Reference

First of all, add OPC Automation wrapper DLL to the reference collection. From the Menu, click [Project] – [Add Reference].

Click, [Reference] button, and then add "OPCDAAuto.dll".

This DLL is included in ORiN2¥CaoOPC¥Bin.

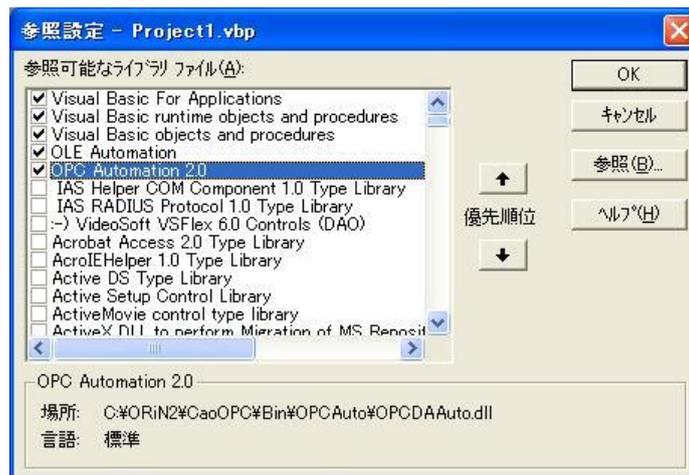


Figure 2-4 Addition of reference

(2) Coding

Create a form which includes one text box. Program the following source codes.

List 2-1	Form1.frm
----------	-----------

```

Dim WithEvents objMyOpcGroup As OPCGroup

Private Sub Form_Load()
    Dim myOpcServer As OPCServer
    Dim myOpcItems As OPCItems
    Dim myOpcGroups As OPCGroups
    Dim myOpcGroup As OPCGroup

    ' connect server
    Set myOpcServer = New OPCServer
    myOpcServer.Connect "OPC.CAO" ' ProgID of CaoOPC is fixed to "OPC.CAO".

    ' add group
    Set myOpcGroups = myOpcServer.OPCGroups
    Set myOpcGroup = myOpcGroups.Add("MyGroup")
    myOpcGroup.UpdateRate = 10 'Set the update interval to 10ms.
    Set myOpcItems = myOpcGroup.OPCItems
    Dim ItemServerHandles() As Long
    Dim ItemServerErrors() As Long
    Dim RequestedDataTypes(1) As Integer
    Dim AccessPaths(1) As String
    Dim ClientHandles(1) As Long
    Dim OPCItemIDs(1) As String

    ' add item
    OPCItemIDs(1) = "ItemA" 'Set the Item name.
    ClientHandles(1) = 1
    RequestedDataTypes(1) = vbDouble
    AccessPaths(1) = "DS1" ' Select from controller list.
    myOpcItems.AddItem 1, OPCItemIDs, ClientHandles, ItemServerHandles, _
        ItemServerErrors, RequestedDataTypes, AccessPaths
    myOpcGroup.IsActive = True 'Start reading.
    myOpcGroup.IsSubscribed = True ' Enable the Event.
    Set objMyOpcGroup = myOpcGroup

```

```
End Sub

Private Sub objMyOpcGroup_DataChange(ByVal TransactionID As Long, _
    ByVal NumItems As Long, ClientHandles() As Long, _
    ItemValues() As Variant, Qualities() As Long, _
    TimeStamps() As Date)
    Text1.Text = ItemValues(1)      ' Updated whenever the value of "ItemA" changes.
End Sub
```



Once this program runs, an event is generated whenever Item's value changes and the value in the text box changes.

3. CaoOPCConfig

3.1. Overview

You can start CaoOPC-dedicated CaoSQLConfig (henceforth CaoOPCConfig) by executing CaoSQLConfig with an argument "OPC" ("CaoSQLConfig.exe OPC"). CaoOPCConfig is a tool that configures CaoOPC parameters for computer, parameters for creating CAO controller objects, and CAO variable object. Information set here is recorded in the registry and the csq file, and read out at the time of CaoOPC startup.

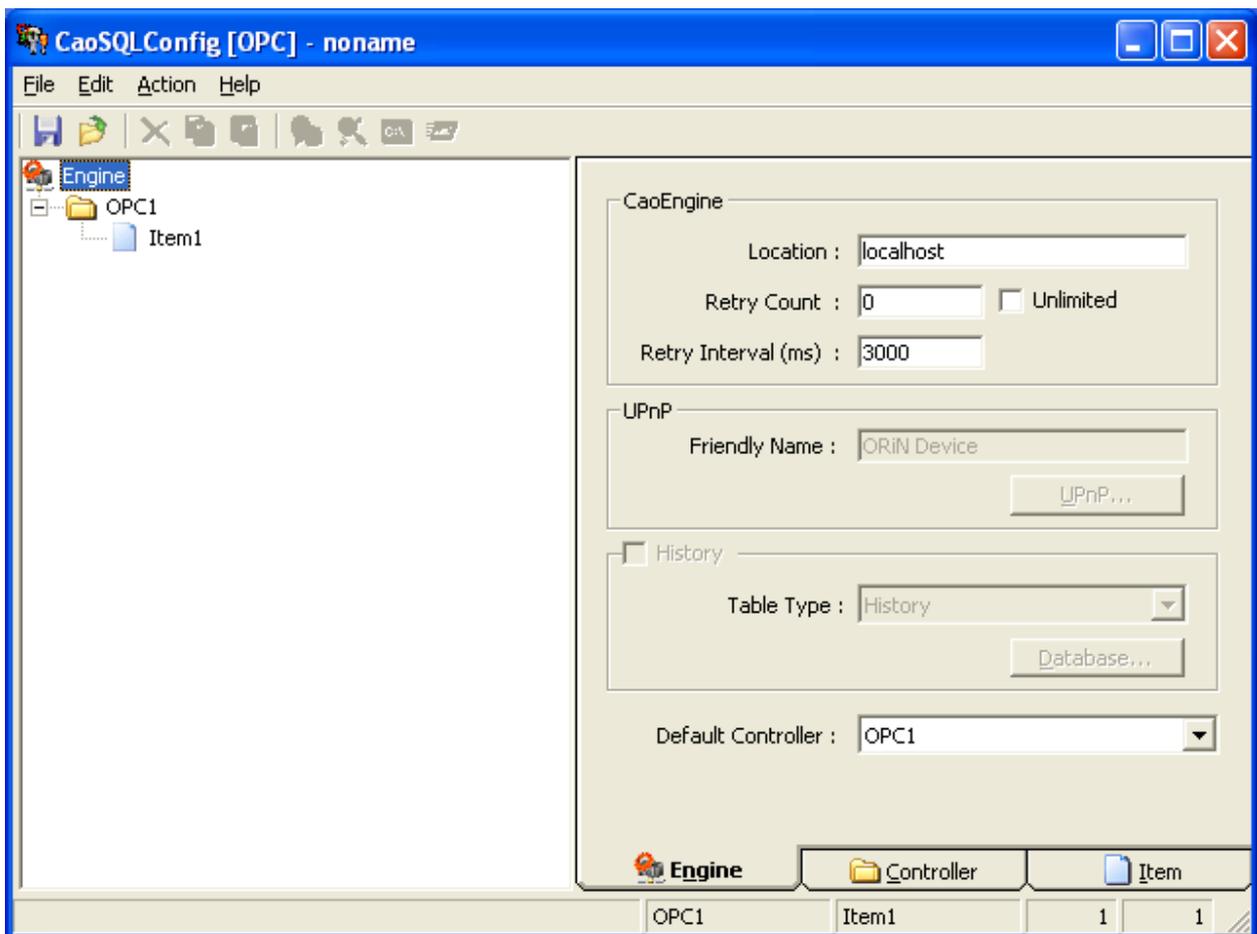


Figure 3-1 CaoOPCConfig screen

3.2. Operation method

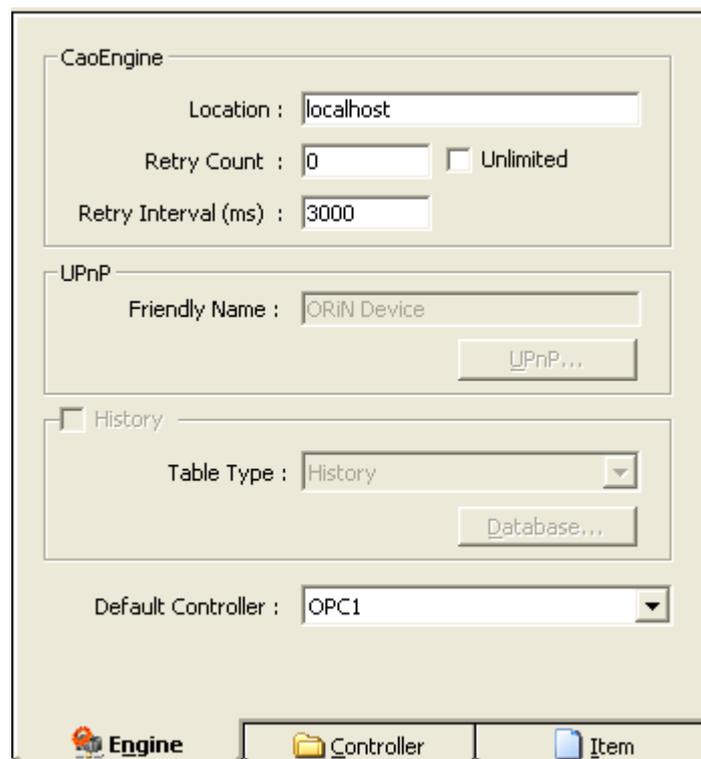
3.2.1. Tab input

There are three tabs on the right bottom of the window. The tab displays information of the object corresponding to the selected node on the tree view. For example, if an Engine node is selected, the tab displays information of Engine.

Because the consistency of input data is not checked, an error may occur at the time of CaoOPC startup, as a result, it could be removed from the list of CaoOPC. To avoid that, check that the data is properly input when you enter data with CaoOPCConfig.

3.2.1.1. Engine tab

Figure 3-2 shows the items on Engine tab. You can set information of Engine as you like. For CaoOPC workspace name, the default name is automatically adopted. Only the existing controller name can be selected for the default controller name.



The screenshot shows the 'CaoEngine' configuration window. It has a title bar 'CaoEngine' and a light beige background. The window is divided into several sections:

- CaoEngine**: Contains three input fields: 'Location : localhost', 'Retry Count : 0' (with an 'Unlimited' checkbox), and 'Retry Interval (ms) : 3000'.
- UPnP**: Contains a 'Friendly Name : ORiN Device' field and a 'UPnP...' button.
- History**: Contains a 'Table Type : History' dropdown menu and a 'Database...' button.
- Default Controller**: Contains a dropdown menu showing 'OPC1'.

At the bottom of the window, there are three tabs: 'Engine' (selected), 'Controller', and 'Item'.

Figure 3-2 Engine tab

You can set the following items in the Engine tab.

[Location]

Enter the start computer name of CAO.exe. If there is no input, CaoOPC starts on the local computer.

[Retry Count, Unlimited]

If “Unlimited” check box is selected, a computer will re-open CaoController until it succeeds if the computer fails to open. Please enter the retry count to “Retry count” or select “Unlimited”.

[Retry Interval]

Specify how long it takes from the failure of the CaoController opening to the re-entry by milliseconds.

[Default Controller]

This controller name will be used if access path is not specified.

3.2.1.2. Controller tab

Figure 3-3 shows the items of Controller tab.

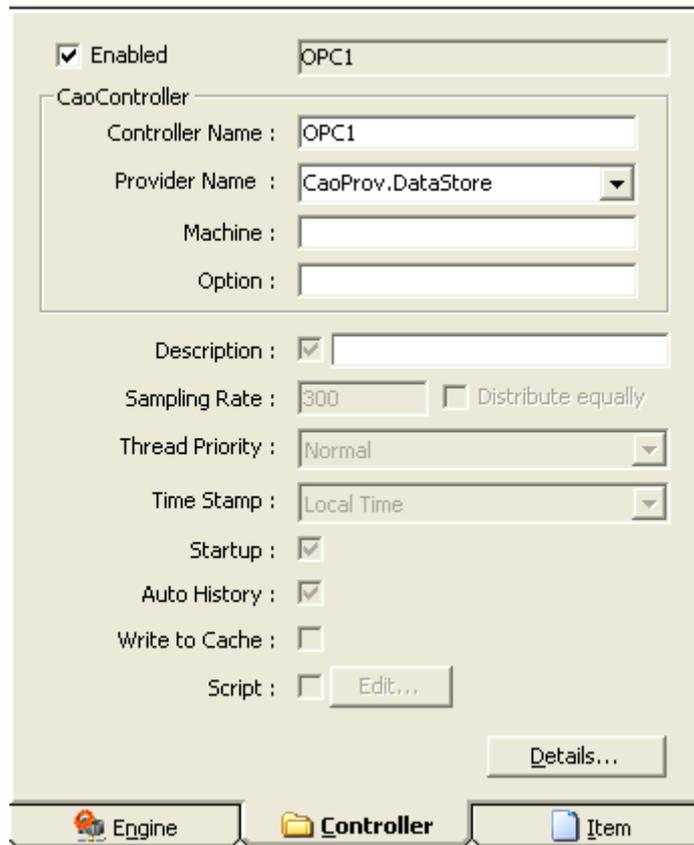


Figure 3-3 Controller tab

You can set information of Controller as you like; however, only "CaoOPC controller name" needs to have a unique information within the registered information. To change "CaoOPC controller name", from the menu bar, click [Edit], select [Rename], and then enter a name. Any other methods are not allowed to change the name. Controller name is not case-sensitive and a name that consists of digit only is not allowed. Symbols of ¥, \$, #, :, !, and / cannot be used.

You can set the following items in the Controller tab.

[Enabled]

Enable/Disable this controller. If this item is set to "Disable", this controller will not be started when CaoSQL starts.

[Controller Name]

Set the name of CaoOPC controller.

This controller name is used as AccessPath.

[Controller Name, Provider Name, Machine, Option]

These parameters are passed when AddController. Values entered here will be used as-is.

[Description]

Clicking this button will display the detailed information of the controller. It is suitable to set non-volatile information because it is registered from XML file.

3.2.1.3. Item tab

Figure 3-4 shows the items of Item tab.

Just like the Controller tab, you can change "Item name" only from the menu bar. (From the menu bar, click [Edit] – [Rename]). Item name is not case-sensitive and a name that consists of digit only is not allowed. Symbols of ¥, \$, #, :, !, and / cannot be used.

The screenshot shows the 'Item' configuration tab. It includes the following fields and controls:

- Enabled: Item1
- Deactivate (Stop auto-sampling)
- CaoVariable
 - Variable Name: Item1
 - Option: [Empty]
 - Class: Controller Class
 - Object Name: [Empty]
- Description: [Empty]
- Attribute: Read / Write
- Link to Item: [Empty]
- Initialize: [Empty] VT_I2
- Force to change: n/a
- Auto History:
- OnChange event:

Buttons: Details... (top right), Engine, Controller, Item (bottom tabs)

Figure 3-4 Item tab

You can set the following items in the Item tab.

[Enabled]

Enable/Disable an item. If an item is set to "Disable", sampling is not performed for the item.

[Item Name]

The item name is set. This item name is used as ItemID.

[Variable Name]

Specify the variable object name of CAO. This setting is used when CaoVariable object is obtained (ICaoController::GetVariable). The input value is used as-is. This is case-sensitive.

[Option]

Specify an option that you add at CaoVariable object creation. This setting is used when CaoVariable object is obtained (ICaoController::GetVariable).

[Class]

Select the type of Variable class. This setting is used when CaoVariable object is obtained (ICaoController::GetVariable).

[Attribute]

Set the attribute of the item..

[Initialization]

Set initial values for items. It is not quite necessary for actual computer though, it might be useful when you set DataStore provider to controller. This setting applies to an item whose attribute is "Read" when CaoVariable object is obtained (ICaoController::GetVariable).

[Description]

Specify the detailed information on the item. Information set here is registered in XML file, so it is suitable for setting nonvolatile information. This setting is used when CaoVariable object is obtained (ICaoController::GetVariable).

[Item Details]

Item Details [OPC1 - Item1]

Data Filtering

Deadband : 0 - 0

Min-Max : 100 - 200

Chattering : 0

Allowed Value

Min-Max-Step : 0 - 0 - 0

List : _____

Default Value : _____

Enable VT_EMPTY :

Data Conversion

Masking : 0

BCD :

Calculation : (NOP) _____

(NOP) _____

Request Type (R) : VT_VARIANT

Request Type (W) : VT_VARIANT

Misc.

Data Type : VT_BOOL

BLOB : 0 _____

EU Unit : _____

Compare Arrays :

Extract Linking :

Linked Property : 0 : Value

OK

Figure 3-5 Item details setting dialogs**[Min-Max]**

Enable/Disable the setting of Low EU(VT_R4) and High EU(VT_R4).

Low EU and High EU are used in the following function to judge the value change.

$$| (\text{last value} - \text{current value}) | > (\text{Deadband percent}^9 / 100) * (\text{HighEU} - \text{LowEU})$$

⁹ Deadband Percent is a property value of OPCGroup.

[Data Type]

Set the native type (Canonical DataType) of the item. Table 3-1 shows the available data types.

Table 3-1 Data type of Item

Data type	Byte	Description
VT_BOOL	1	1: True, Other than 1: False.
VT_I1	1	Integer signed
VT_UI1	1	Integer unsigned
VT_I2	2	Single-precision Integer (16 bits), signed
VT_UI2	2	Single-precision Integer (16 bits), unsigned
VT_I4	4	Double-precision Integer (32 bits), signed
VT_UI4	4	Double-precision Integer (32 bits), unsigned
VT_R4	4	Single-precision Floating point (32 bits)
VT_R8	8	Double-precision Floating point (64 bits)
VT_CY	8	Currency type Fixed point number with four digit after the decimal point. It is stored as an 8-byte signed integer scaled by 10000.
VT_BSTR	Variable	Character string After the number of characters expressed by VT_UI4, UNICODE character and NULL terminator are added.
VT_DATE	8	Same as VT_R8. Total date from 1899/12/30.
VT_VARIANT	Variable	VARIANT type

[BLOB]

Set the size (VT_I4) of BLOB and BLOB character string (VT_BSTR).

[EU Unit]

Set the value (VT_BSTR) of EU Unit.

3.2.2. Menu

A basic operation in CaoOPCConfig is selected from the Menu bar. You can select some menus by right-clicking the tree view though, the menus available are exactly the same as the ones opened from the Menu bar.

3.2.2.1. File menu

Save and load the created data.

Extnesion is different depending on the operation.

For [Open], [Save], and [Save As...], use "csq".

For [Import] and [Export], use "csx".

For "Import from CSV file" and "Export to CSV file", use "csv".

[New]

Create a new csq file.

[Open]

Read the setting information from the specified csq file. ¹⁰

When the reading is canceled, the reading result is not reflected.

[Save]

Write the current settings over a csq file.

[Save As...]

Save the current settings in a csq file.

[Import]

From a csx file, import data to the currently selected node.

[Export]

Export the currently selected node to a csx file.

[Import from CSV file]

From a csv file, import data to the currently selected node.

[Export to CSV file]

¹⁰ If Windows 9X/Me is used, please note that data loss might occurs if the size of read data is 64KB or larger because of OS restrictions

Export the currently selected node to a csv file.

[Exit]

Close CaoOPCConfig program.

3.2.2.2. Edit menu

Users can edit items, such as adding, deleting, naming of items, from this menu.

[Add Controller]

Add a controller that you want to read from the CaoOPC. The same controller name as the existing controller cannot be added. (This is not case-sensitive.) Once a valid name is entered, it will be added to the tree. The detailed information is selected from the tab on the right of the window.

[Add Item]

Add an item to the currently selected node of the tree view. The same item name as the item name that exists in the selected controller cannot be added. (This is not case-sensitive.) Once a valid name is entered, it will be added to the tree. The detailed information is selected from the tab on the right of the window.

[Rename]

Rename the controller/item that is currently selected from the tree view. Already registered controller/item name cannot be changed. (This is not case-sensitive.)

[Delete]

Delete the node selected from the tree view. When a controller is deleted, all items registered in the controller are recurrently deleted. Or, you can do the same process by pressing the Delete key on the keyboard while the item is selected on the tree

[Copy]

Copy the selected node from the tree view. If the controller is copied, registered items are copied as well. Note that the trigger information is not copied. "Paste" will add a clone of the copied information to the node.

[Paste]

Paste the copied information to the node with a new name.

[Sort]

Sort the node of the selected level by ascending order.

3.2.2.3. Action menu

Action menu provides the CaoOPC.exe-related settings, such as language, log, security and others. Setting will be stored in the registry.

[Option]

Display the CaoOPC option setting dialog.

[General]

The General tab includes CaoOPC general settings, such as process priority setting and the language setting.

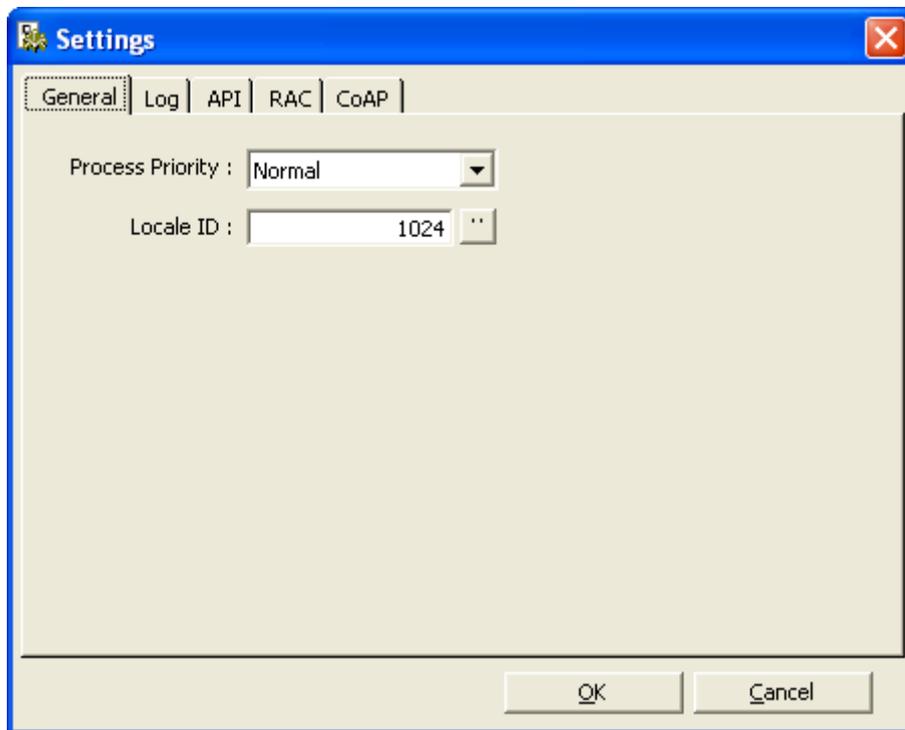


Figure 3-6 CaoOPC environmental setting (General)

[Process Priority]

Set the process priority of CaoOPC. The order of the priority is as follows.

Real time > High > **Normal** > Idle

[Locale ID]

Select a language ID you use.

[Log]

The Log tab includes CaoOPC log output-related settings.

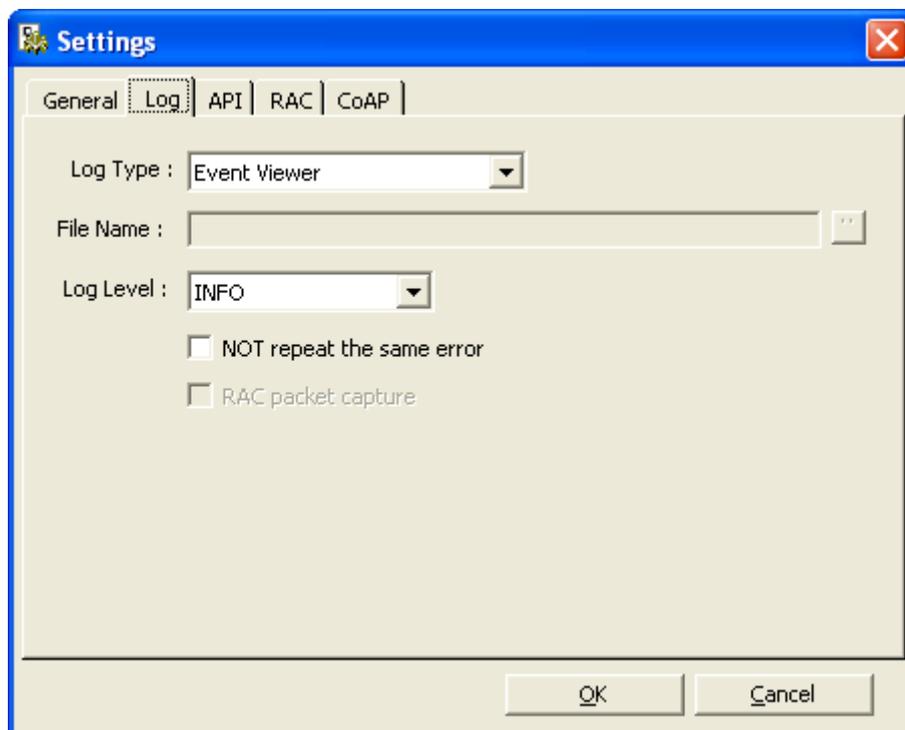


Figure 3-7 CaoOPC environmental setting (Log)

[Log Type]

Select the output type of the log of CaoOPC.exe.

Table 3-2 lists available log type for output.

Table 3-2 Log type

Output to	Description
Console	Output to the console
Message box	Output to the message box (At the timing of service start)
Event viewer	Output to the event viewer (At the timing of service start)
Debugging viewer	Output debugging
Text file	Output to the specified text file

[File Name]

Specify the file path to output the log. This path is used when the setting of Log Type is "Text file".

[Log Level]

Set the log output level. The following five levels are prepared. "FATAL" is the highest severity level whereas "DEBUG" is the lowest. The default setting is "INFO".

FATAL > ERROR > WARN > **INFO** > DEBUG

[NOT repeat the same error]

Inhibit recurring same error output. If this checkbox is selected, the error message caused by the repeatedly occurring error is issued only once.

[API]

No items are available.

[RAC]

No items are available.

3.2.2.4. Help menu

This menu includes helps and license registration menu.

[Version]

Version information is displayed.

3.3. Registry structure information

The registry used is the same as CaoSQLConfig. For details, refer to "4.3 Registry composition information" of '[CaoSQL user's guide](#)'.

4. OPC provider

4.1. Overview

This chapter briefly explains "CAO provider for OPC connection" (hereafter, OPC provider) that establishes a connection with PLC (Programmable Logic Controller) through CAO from OPC (OLE for Process Control) server. For details about OPC provider, refer to "[OPC provider user's guide](#)".

This provider can access PLC and the indicator panel with not only the robot but also the OPC server the tool for CAO as well as the robot.

This OPC provider enables CaoVariable object to refer to an Item's value on a OPC server.

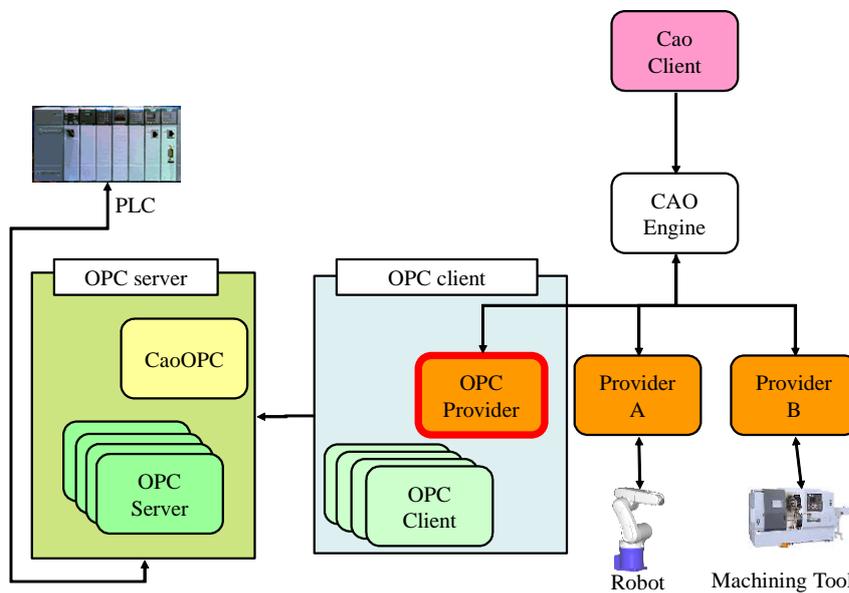


Figure 4-1 CAO provider for OPC connection

4.2. Method property

4.2.1. CaoWorkspace::AddController method

The following shows the specifications of AddController method for CaoWorkspace class of CAO.

```
AddController
(
    "<Controller name>" ,           // Controller name
    "CaoProv. OPC " ,              // Provider name. Fixed.
    "<Machine name>" ,             // Machine name that the provider runs
    "<Option>"                      // Option character string (OPC option)
)
```

In this example, Provider Name is fixed. <Machine Name> is a computer name that the provider runs.

For about the format of <Option>, refer to "[OPC provider user's guide](#)".

The following shows sample code.

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

4.2.2. CaoController::AddVariable method

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

```
CaoController:: AddVariable
(
    "<Variable identifier>" ,      // Item ID.
    "<Option>"                      // Option character string (OPC option)
)
```

To use OPC provider, set the argument as follows.

< Variable name >::=< Item ID >

< Option > ::= [[AccessPath=<access path>] [, ActiveState=<0|1>] [,RequestType=<Variable type of reception>] [,Source=<data source>]]

For about the format of <option>, refer to "[OPC provider user's guide](#)".

The shows the sample code.

```
AddVariable
(
    "Item1",                       // Item ID is Item1.
    "AccessPath=OPCC1, ActiveState=0, RequestType=8, Source=1" // Access path is OPCC1,
```

);

Active state is InActive,
Variable type for reception is VT_BSTR,
Data source is OPC_DS_CACHE