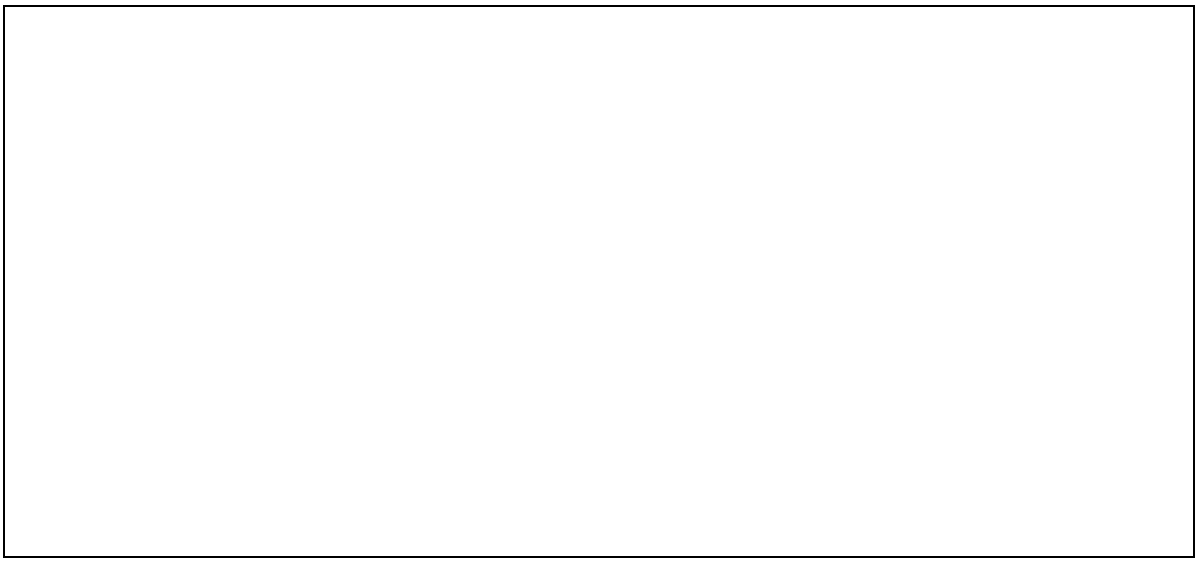
CaoSQL Tools

User's guide

Version 1.8.0

December 14, 2013



[Revision history]

Date	Version	Content
2005-07-06	1.0.0	First edition.
2011-02-22	1.1.0	Add special controller name information to CaoSQL provider
2013-12-14	1.8.0	Add Execute command, ID property

Contents

1. Introduction

This document is a user's guide of CaoSQL[1], -dedicated tools.

This document describes CaoSQLTester, CaoSQLLauncher, CaoSQLCmd, CaoSQLProvider, LinkViewer, Timing chart add-in tool, and jCaoSQL.

CaoSQLTester is a client application that equips all interfaces used in CaoSQL. This tool allows you to execute CaoSQL methods. Also, you can check the retrieval and setting of property.

CaoSQLLauncher starts CaoSQL.exe on specified computer.

CaoSQLCmd is a console application that sends various command via CaoSQL public interface. With this tool, you can control a controller individually, for example, to stop a certain controller's sampling. Also, you can perform sampling only for a certain period of time by combining this tool and task scheduler, which is accompanied by Windows.

LinkViewer add-in tool is one of the Microsoft Excel add-in tools that displays the link-status of LinkItem property, which is one of CaoSQLItem properties configured by CaoSQLConfig, with visual image. With this tool, users can easily find the entire relationship of items.

Timing chart add-in tool is one of the Microsoft Excel add-in tools that displays multiple CaoSQLItems' values in the timing chart so that users can easily observe on/off timing of signals. This tool enables you to find the turning ON/OFF timing of a certain item easily.

jCaoSQL is a framework to control CaoSQL from java application. With this tool, user can access resources, such as a robot controller, through CaoSQL interface

[1] Please refer to the chapter of "4 CaoSQLConfig" of 'CaoSQL user's guide' for CaoSQLConfig.

2. CaoSQLTester

2.1. Overview

This section describes CaoSQLTester that offers users to check CaoSQL performance and behavior. CaoSQLTester implements methods and properties of CaoSQLEngine, CaoSQLController, CaoSQLItem, and CaoSQLHistory, so user can test these items.

2.2. Operation method

There is an object tree on the left pane. When you double-click the node on the tree view, a window corresponding to the double-clicked node will be displayed on the right of the tree. For example, when you double-click CaoSQLEngine node, CaoSQLEngine window will be displayed.

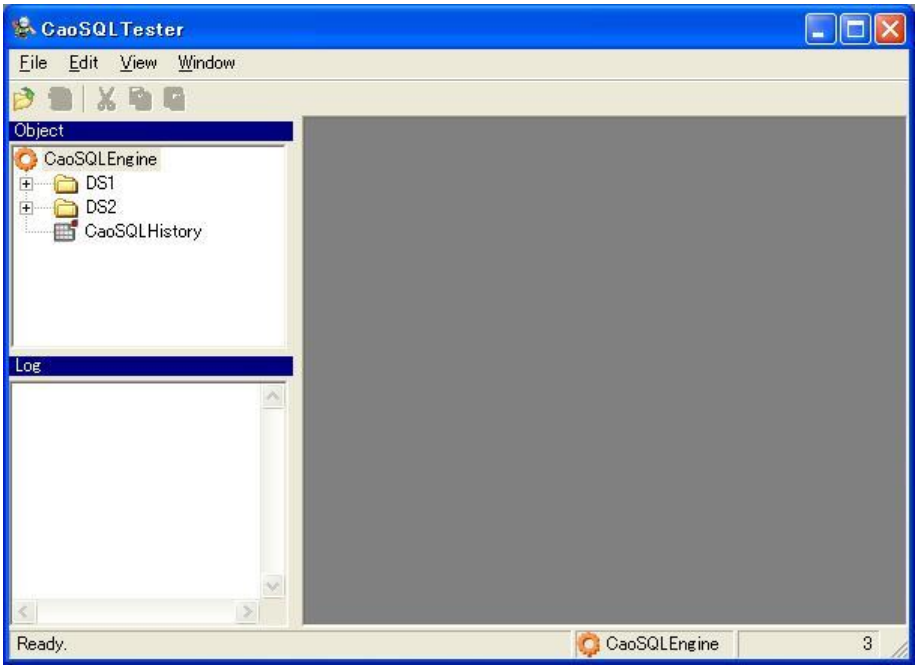


Figure2-1 CaoSQLTester start window

2.2.1. CaoSQLEngine

CaoSQLEngine window allows you to execute CaoSQL Engine's functions

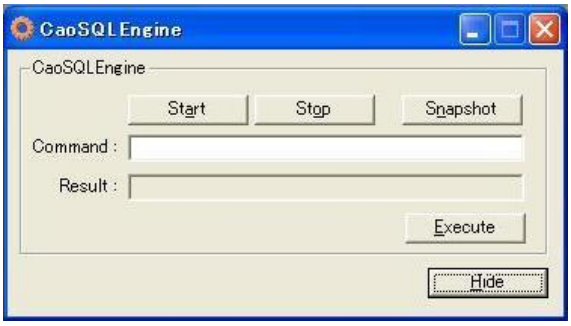


Figure2-2 CaoSQL Engine window

[CaoSQL Engine sampling start] – Start

Start all controllers' sampling

[CaoSQL Engine sampling stop] – Stop

Stop all controllers' sampling

[CaoSQL Engine snap shot] – Snapshot

Execute sampling for all items in all controllers only one time

[CaoSQL Engine command execute] – Execute[2]

Execute an extended command of CaoSQL Engine

[2] This function has not been implemented

2.2.2. CaoSQLController

CaoSQLController window allows you to execute CaoSQLController's functions.

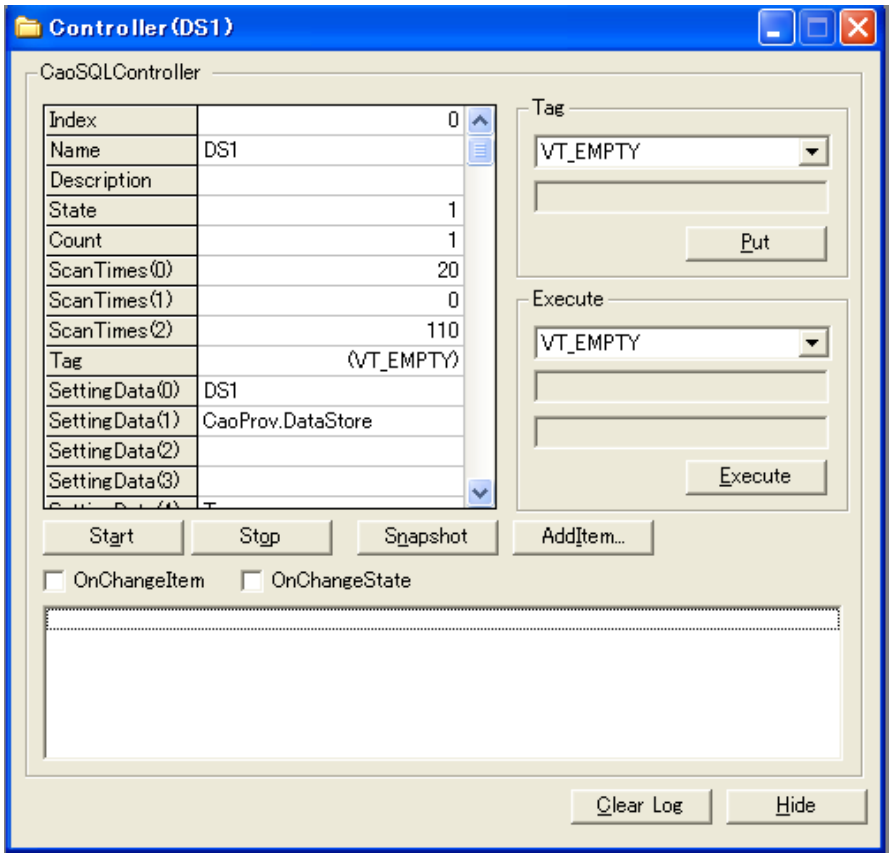


Figure2-3 CaoSQL Controller window

[CaoSQL controller sampling start] – Start

Start sampling of all items under the controller

[CaoSQL controller sampling stop] – Stop

Stop sampling of all items under the controller

[CaoSQL controller sampling] – Snapshot

Execute sampling for all items under the controller for only one time

[CaoSQL controller tag writing] - Put

Write data into a controller tag

Data should be specified by data types designated in RAC[3](Robot Action Command) , which is a Denso's robot language.

The following shows example of data entry.

Example1)	Type:I2	Parameter:100	Value :100
Example2)	Type:BSTR	Parameter: Sample	Value :"Sample"
Example3)	Type:VARIANT	Parameter:(8, Sample)	Value :"Sample"
Example4)	Type:ARRAY|I2	Parameter:100, 200, 300	Value :100, 200, 300
Example5)	Type:ARRAY|VARIANT	Parameter:(8, Sample), (2, 100)	Value :"Sample", 100

[CaoSQL controller command execution] – Execute

Execute a specified command.

The parameter is specified by RAC. エラー! ブックマークが定義されていません。

[CaoSQL controller item change event] – OnChangeItem

Selecting this check box will issue OnChangeItem event when value of the registered item changes. The name and value of the item that issued OnChangeItem event will be displayed in the log window on the bottom.

[CaoSQL controller change of state event] – OnChangeState

Selecting this check box will issue OnChangeState event when a registered controller's state value changes. The controller status value will be displayed in the log window on the bottom.

[CaoSQL controller item addition] – AddItem

Add an item to CaoSQL controller.

Items to be specified are the same as CaoSQLConfig. For details, refer to "4.CaoSQLConfg" of 'CaoSQL user's guide' .

[3] Refer to "3.2 How to write data" of 'RAC user's guide' for details.

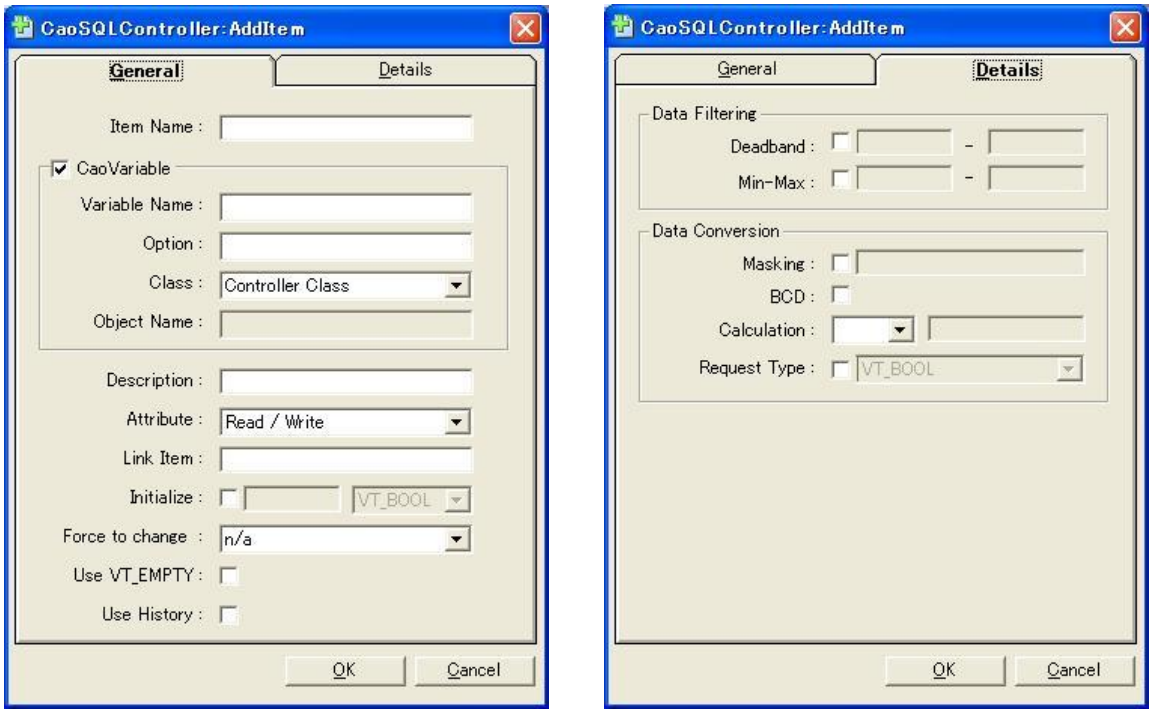


Figure2-4 Adding items

2.2.3. CaoSQLItem

CaoSQL item window allows you to execute CaoSQLItem functions.

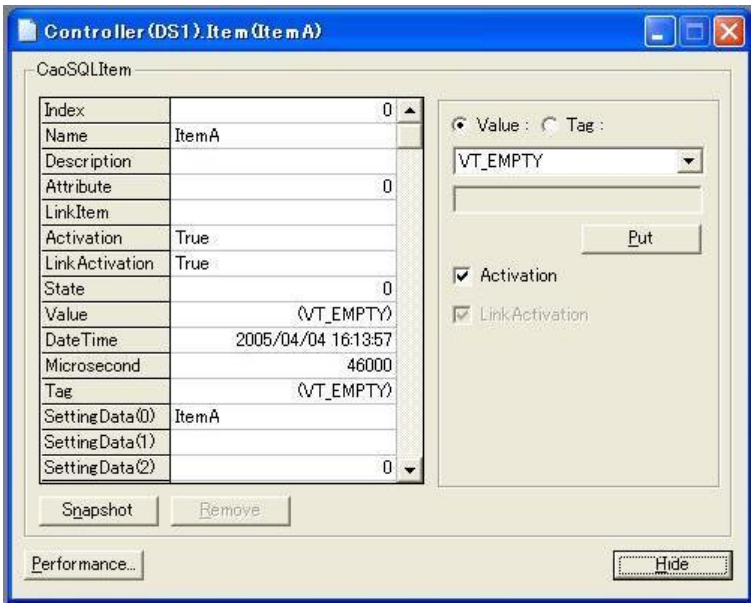


Figure2-5 CaoSQL Item window

[CaoSQL item snap shot] – Snapshot

Execute sampling the selected item only one time

[CaoSQL item deletion] – Remove

Delete an item if the item is dynamically-added item.

This process is available only for dynamically-added items.

[CaoSQL item value/tag] – Value / Tag

Select an output destination of the item from the pull down menu of Property.

[CaoSQL item writing] – Put

Write the contents of Value textbox into item tag or value.

The data is specified by RAC. エラー! ブックマークが定義されていません。

[CaoSQL item activation] – Activation

If this check box is selected, CaoSQLItem is activated.

[CaoSQL item value link activation] – LinkActivation

Selecting Link Activation check box will enable to transfer values to the item's link destination.

[CaoSQL item performance check] – Performance

Check the reading-/writing-performance of item.

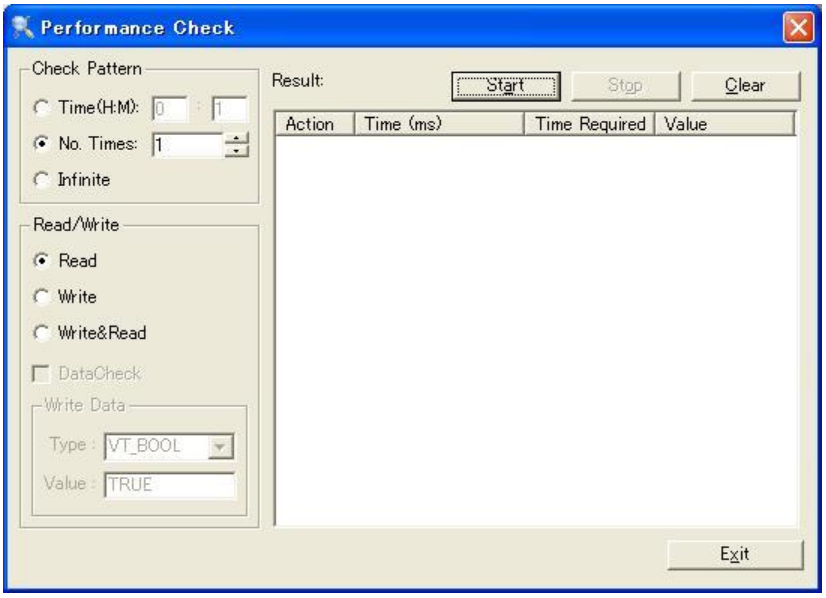


Figure2-6 Performance Check window

[Check pattern] – Time / No. Times / Infinite

Set the time period or the frequency of the performance check with the following setting items.

Time	:	Specify how long performance is checked. Unit is [*hour*]:[*minute*]
No. Times	:	Specify the number of performance check execution.
Infinite	:	Continue performance check until the stop button is clicked.

[Check item] – Read / Write / Write & Read

Select an item that you intend to check the performance.

Read	:	Check the performance of the reading only
Write	:	Check the performance of the writing only
Write & Read	:	Check the performance of both the writing and reading

[Data check] – DataCheck

Check whether the reading data and the writing data is the same. This function is available only when "Write & Read" is selected in the "Check item".

An error will be displayed in a dialog when any discrepancy occurs between the reading data and the

writing data.

[Writing data] – WriteData

Set the data type and value to write. This function is available when "Write" or "Write & Read" is selected in the "Check item".

[Performance check start] – Start

Start the performance check

[Performance check stop] – Stop

Stop the performance check

[Clear log] – Clear

Clear the test result.

2.2.4. CaoSQLHistory

This dialog determines the functions of CaoSQLHistory.

When History is invalid, SQL Command text box, Execute button, and Result grid are not available.

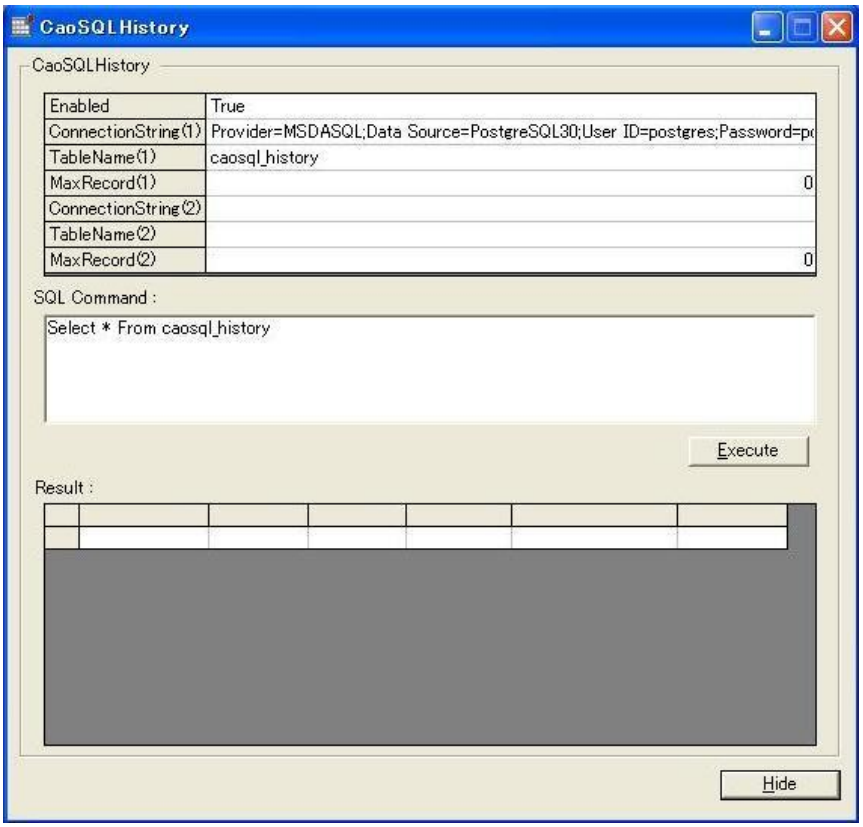


Figure2-7 CaoSQL History window

[CaoSQL history SQL statement entry] – SQL Command

Enter SQL command to execute.

[CaoSQL history SQL statement execution] – Execute

Execute SQL statement written in the SQL Command text box.

[CaoSQL history SQL statement execution result] – Result

Display the result of SQL statement execution in the grid.

3. CaoSQLCmd

3.1. Overview

This section describes CaoSQLCmd.exe that controls CaoSQL from the command line. The purpose of this tool is to manage CaoSQL's behavior on a time basis. CaoSQLCmd is a console application. Using this tool in batch processing from the task manager, which is mounted in the computer's operation systems (Windows NT/ 2000 / XP, etc.), will facilitate the time management of CaoSQL.

Also, synchronizing /Value/put command with UPS-shutdown will inform the shutdown to CapSQL application in a given timing, and this will terminate the application safely.

3.2. Function

Following five commands are available in CaoSQLCmd. These commands can be combined arbitrarily, and can execute various processing for CaoSQL according to the following syntax.

(1) /Start command

```
CaoSQLCmd  /Start [<Target Controller>]
```

(2) /Stop command

```
CaoSQLCmd  /Stop  [<Target Controller>]
```

(3) /Snapshot command

```
CaoSQLCmd  /Snapshot [<Target Controller>[ <Target Item>]]
```

(4) /State command

```
CaoSQLCmd  /State <Target Controller>[ <Target Item>]
```

(5) /Value command

```
CaoSQLCmd  /Value <Target Controller> <Target Item>[ /Put <Data type>, <Value>]
```

Command names and controller names are not case-sensitive. Elements enclosed by square brackets can be omitted. Target controller and/or target item can be omitted. If it is omitted, commands will be executed to all controllers and/or commands. Since all commands are to be executed to CaoSQL which has been executed, it will return an error if CaoSQL is not executed. If you use /Put in /Value command, enclose the value with double-quotations if it contains space. For information about data format, refer to "How to write data" in 'ORiN2 programming guide'

Table3-1 Command Execution Example List of CaoSQLCmd

Command	Example	Description
/Start	CaoSQLCmd.exe /Start	Start all controllers' threads.
	CaoSQLCmd.exe /Start RC1	Start target controller's thread.
/Stop	CaoSQLCmd.exe /Stop	Stop all controllers' threads.
	CaoSQLCmd.exe /Stop RC1	Stop target controller's thread.
/Snapshot	CaoSQLCmd.exe /Snapshot	Take snapshots of all controllers.
	CaoSQLCmd.exe /Snapshot RC1	Take a snapshot of the target controller.
	CaoSQLCmd.exe /Snapshot RC1 ItemA	Take a snapshot of the target controller's specified item.
/State	CaoSQLCmd.exe /State RC1	Obtain target controller's state.
	CaoSQLCmd.exe /State RC1 ItemA	Obtain specified item's state of target controller
/Value	CaoSQLCmd.exe /Value RC1 ItemA	Obtain specified item's value of target controller
/Value /Put	CaoSQLCmd.exe /Value RC1 ItemA /Put 8, Test	Set specified item's value of target controller.

/Start

Start command executes Start method of CaoSQLEngine/CaoSQLController interface. In this command, you can specify a controller name to start in argument.

/Stop

Stop command executes Stop method of CaoSQLEngine/CaoSQLController interface. In this command, you can specify a controller name to stop in argument.

/Snapshot

Snapshot command executes Snapshot method of CaoSQLEngine/CaoSQLController/CaoSQLItem interface. You can specify target controller name and target item name

/State

State command obtains State property of CaoSQLController/CaoSQLItem interface. You can specify target controller name and target item name.

/Value

Value command executes get_Value method of CaoSQLItem interface. By adding “/put” option, put_Value method is executed. For about data format specified by “/put” option, refer to "2.2.5. How to write data" of 'ORiN2 programming guide'. Please note that the syntax of Value command is slightly different from put_Value method.

One example of this command usage is to inform the shutdown of the application. Synchronize this command with UPS-shutdown, and then program such a manner that arbitral item’s value changes when shutdown. This will inform shutdown to the application by means of OnChangeItem event.

4. CaoSQLLauncher

4.1. Overview

CaoSQLLauncher starts CaoSQL.exe on specified computer. Registering a shortcut icon of this launcher on Startup menu will allow only limited users to initiate CaoSQL. This shortcut can be placed on the task tray as well.

4.2. Operation method

Figure4-1 CaoSQL Launcher main window

[Hostname] – Host

Specify computer name or IP address to start where CaoSQL.exe is stored.

[CaoSQL start] – Start

Start CaoSQL.exe on the specified host

[CaoSQL stop] – Stop

Stop CaoSQL.exe.

4.3. Command line argument

In CaoSQLLauncher, the following command line arguments are provided.

```
CaoSQLLauncher.exe[ HIDE][ START[ <Hostname>]]
```

HIDE	:	Start CaoSQL.exe with CaoSQLLauncher being stored in the task tray
START	:	Start CaoSQL.exe
<Hostname>	:	Set the hostname

Example1) Start default setting's CaoSQL.exe (localhost) with CaoSQLLauncher being stored in the task tray.

```
CaoSQLLauncher.exe HIDE START
```

Example2) Start CaoSQL on the TestServer.

```
CaoSQLLauncher.exe START TestServer
```

5. CaoSQL provider

5.1. Introduction

This section explains CaoSQL provider that is one of the CAO providers. CaoSQL provider has been developed to utilize CaoSQL, which is a client tool of CAO. CaoSQL provider obtains data that CaoSQL continuously collects through CAO interface. CaoSQL provider will expand CaoSQL structure in distributed computing environment.

5.2. Overview

CaoSQL provider internally interprets CaoSQL interface to CAO interface based on information that is registered by CaoSQLCofig tool. That is to say, user can create system structure as if two CaoSQLs are cascaded. (See Figure5-1).

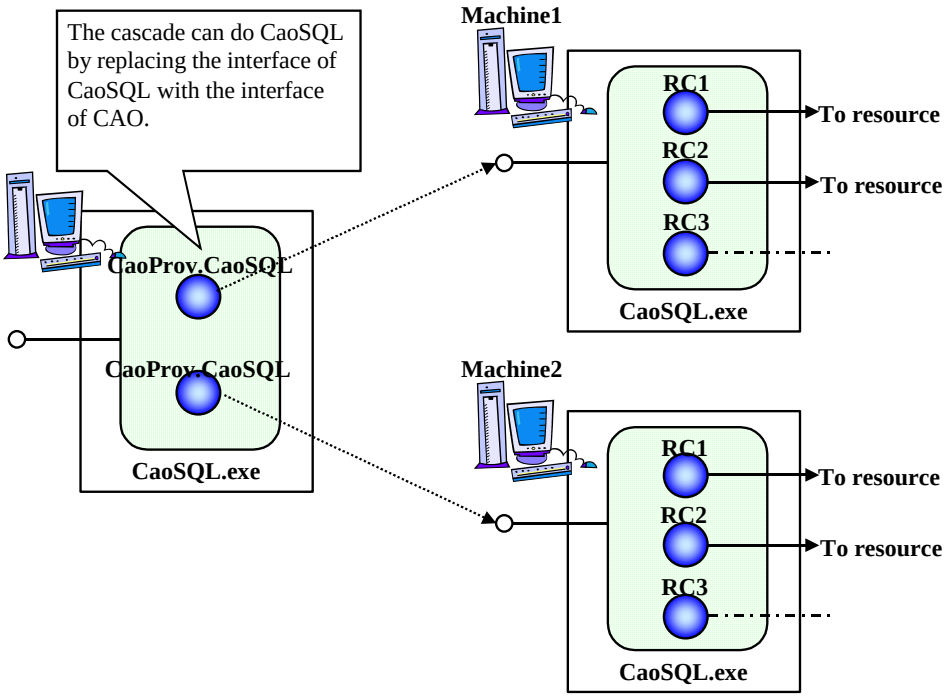


Figure5-1 Cascade of CaoSQL

5.3. How to use CaoSQL provider (CaoProvCAOSQL.dll.)

5.3.1. Introduction

CaoSQL provider (CaoProvCAOSQL.dll) is a CAO provider that accesses providers via CAO interface. This provider implements Controller class and Variable class and these classes are used when CAO accesses CaoSQL provider

Table5-1 CaoProvCAOSQL.dll information

File name	CaoProvCAOSQL.dll
ProgID	CaoProv.DNWA.CAOSQL
Registry	regsvr32 CaoProvCAOSQL.dll
Delete the registry	regsvr32 /u CaoProvCAOSQL.dll

CaoSQL provider offers the following functions in addition to that of provider template offers: obtaining controller object, executing command from controller, obtaining variable object, obtaining variable attribute, obtaining and setting variable value. For details, see Table 5-2

Table5-2 List of available methods and properties in CaoSQL provider

Object name	Method/Property
CaoWorkspace	AddController
CaoController	AddVariable
	Execute
	get_ID
	get_VariableNames
CaoVariable	get_Value
	put_Value
	get_Attribute
	get_DateTime
	get_ID
	get_Microsecond

5.3.2. CaoWorkspace::AddController method

CaoSQL provider creates CaoSQLEngine and CaoSQLController object when Controller object is created

Table5-3 Option character string of CaoWorkspace::AddController

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Controller
=< CaoSQL controller name ></td><td>Controller name registered in CaoSQL
(default: Controller name of provider)
The following reserved names can be used as a controller name.

<table><tr><th>Controller name</th><th>Details</th></tr><tr><td>$ENGINE$</td><td>Use this controller name when you obtain the controller name list registered in CaoSQL by using CaoController::get_VariableNames.
You can also execute CaoSQLEngine::Execute by using CaoController::Execute.
CaoController object whose controller name is $ENGINE$ can execute get_VariableNames and Execute only.</td></tr><tr><td>$HISTORY$</td><td>With this controller name, you can execute SQL for CaoSQL History database by using CaoController::Execute method.
If you create a CaoController object whose name is $HISTORY$, only Execute method is available.</td></tr></table></td></tr>
<tr><td>Machine=<computer name ></td><td>Computer name where CaoSQL is running, or computer name where you intend to run CaoSQL.(default: Computer name where provider is executed)</td></tr>
<tr><td>ConvToArray=True/False</td><td>Enable conversion of ADO Recordset object into VARIANT array (default: False). This option is available when the controller name is “$HISTORY$".</td></tr>
</table>

```
AddController
(
    "<Controller name>",                // Controller name of provider
    "CaoProv.DNWA.CAOSQL",              // Provider name (Fixed)
    "[<Machine name>]",                 // Computer name where the provider is executed
    "[[Controller=<CaoSQLController name>][, Computer=<Computer name>]]"
                    // Registered CaoSQL controller-name,   CaoSQL execution machine name
)
```

(Example)

```
AddController
```

```
(
    "CSQsCtrl1",                    // Use Ctrl1 for controller-name of the provider
    "CaoProv.DNWA.CAOSQL",          // Provider name (Fixed)
    "",                             // Execute provider with a local computer. (in-process)
    "Controller=RC1, Machine=machine1"
                // Use registered controller "RC1". Connect with CaoSQL of "computer 1".
)
```

5.3.3. CaoController::AddVariable method

CaoSQL provider creates a CaoSQLItem object when a Variable object is created. Since variable name specified with AddVariable is used as CaoSQLItem name,. enter CaoSQL item name you want to obtain.

```
AddVariable
(
    "<CaoSQLVariable name>"                 // Variable name of provider=CaoSQLItem Name
)
```

5.3.4. CaoController::Execute method

The execution result of this method differs depending on the controller name specified by AddController.

(1) Controller name is $ENGINE$

Execute method of CaoSQLEngine is executed. Command character string specified in arguments is handed to CaoSQLEngine::Execute as-is. Parameter character string of the argument is not used. The execution result of "CaoSQLEngine::Execute" is returned as the result of this method; however, in the following command, each process will be executed.

Command name	Argument	Details
$CSQP_GET_COUNT$	none	Number of CaoSQLController object (get_Count)

(2) Controller name is $HISTORY$

Execute method of CaoSQLHistory is executed. Command character string specified in argument is handed to CaoSQLHistory::Execute as-is. Parameter character string of the argument is not used. The execution result of this method differs depending on the value of ConvToArray option at AddController timing.

・ ConvToArray = False

Execution result of CaoSQLHistory::Execute is returned as ADO Recordset object.

・ ConvToArray = True

Execution result of CaoSQLHistory::Execute is returned as VARIANT array. The first element of VARIANT array stores the field name list of execution result. The succeeding elements store the records of execution results in order.

(3) Controller name is other than above

Execute method of CaoSQLController is executed. Command character strings and parameter strings specified in arguments are directly passed to CaoSQLController::Execute. Execution result of CaoSQLController::Execute is returned as the result of this method; however, note that the following commands are processed distinctively.

Command name	Argument	Details
$CSQP_GET_COUNT$	none	Number of CaoSQLItem object (get_Count)
$CSQP_GET_STATE$	none	State of CaoSQLContoller (get_State)

5.3.5. CaoController::get_Help property

Obtain Description property of CaoSQL controller.

5.3.6. CaoController::get_ID property

Obtain Index property of CaoSQL controller.

5.3.7. CaoController::get_Tag property

Obtain Tag property of CaoSQL controller.

5.3.8. CaoController::put_Tag property

Set Tag property of CaoSQL controller.

5.3.9. CaoController::get_VariableNames property

If the controller name is "$ENGINE$", the list of CaoSQLController registered in CaoSQL is obtained.

If the controller name is other than "$ENGINE$", the list of CaoSQLItem under CaoSQLController specified with AddController is obtained..

5.3.10. CaoVariable::get_Attribute property

Obtain the attribute value of CaoSQL item. Table 5-4 shows the list of attribute values obtained.

Table5-4 Attribute value that can be obtained

Attribute value	Description
0	Read/Write
1	ReadOnly
2	WriteOnly

5.3.11. CaoVariable::get_DateTime property

Obtain the date of CaoSQL item update.

5.3.12. CaoVariable::get_Help property

Obtain Descripton property of CaoSQL item.

5.3.13. CaoVariable::get_ID property

Obtain CaoSQL item ID.

5.3.14. CaoVariable::put_ID property

Set CaoSQL item ID.

5.3.15. CaoVariable::get_Microsecond property

Obtain the microsecond of CaoSQL item updated date.

5.3.16. CaoVariable::get_Tag property

Obtain the CaoSQL item Tag.

5.3.17. CaoVariable::put_Tag property

Set the CaoSQL item Tag.

5.3.18. CaoVariable::get_Value property

Obtain CaoSQL item value.

5.3.19. CaoVariable::put_Value property

Set CaoSQL item value.

6. Link viewer

6.1. Overview

Link viewer[4] displays the link status based on the LinkItem data configured by CaoSQLConfig.

Link viewer runs as a Microsoft Excel add-in. For information about add-in setup, see 6.2.

With this Link viewer, you can easily check the link status of all items even if one item links with multiple items. In addition, the Link viewer displays number of links as well as Description that are configured by CaoSQLConfig.

If a newly created sheet name is already exist in the file, Link viewer overwrites the existing file. To avoid overwriting, copy the sheet and save it as a different name before creating a new sheet.

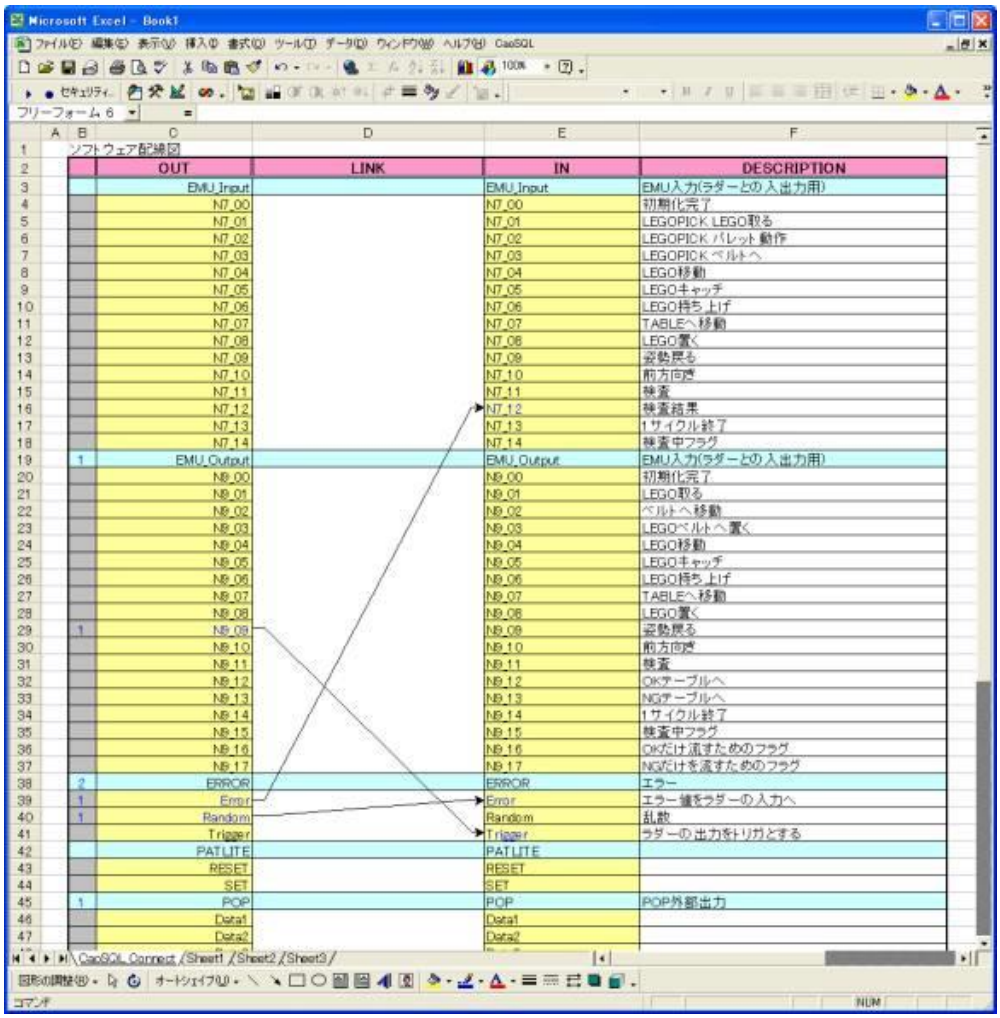


Figure 6-1 Link viewer screen

6.2. Registering Add-in file

Before using Link viewer, you need to register add-in file (CaoSQL.xla) so that Link viewer runs as an

[4]This tool uses a part of the result of the research that JSPMI (Japan Society for the Promotion of Machine Industry) executed with the assistance of the cycle race. This tool is open to the public to the support of the source though it is out of the support coverage.

add-in of Microsoft Excel. Excel add-in is usually stored in the following folder according to the purpose.

Common to All user	:	[Office installation directory]¥Office¥Library
Each user	:	[Profile directory according to user]¥Application Data¥Microsoft¥AddIns

For instance, users who use Windows XP store the file under the following destination:

C:¥Documents and Settings¥<Each user's folder>¥Application Data¥Microsoft¥AddIns

Figure6-2 shows a folder path on Windows XP. However, please note that the drive path to AddIns folder may change according to the environment[5].

Instead of the process above, you can register add-in file by specifying it in Add-in dialog box.

Figure6-2 Add-Ins folder (each user)

[5]This example has been executed in the following environments.
OS: Windows XP, Drive at installation destination of OS : C drive, Office : Microsoft Office2000

When the process mentioned above has been completed, CaoSQL will be displayed in the list of Excel Add-in (Figure6-3).

Figure 6-3 Selecting Add-in

From the Add-in list, select CaoSQL checkbox, and then click OK. Add-in is properly added when CaoSQL is displayed in the main menu of Excel (Figure 6-4). Add-in execution is described in the next chapter.

Executing Link viewer

When CaoSQL is added, To display Link Viewer in a new sheet automatically, from the menu bar, click CaoSQL, click Execute Link Viewer. The link viewer is automatically displayed in the newly created sheet.

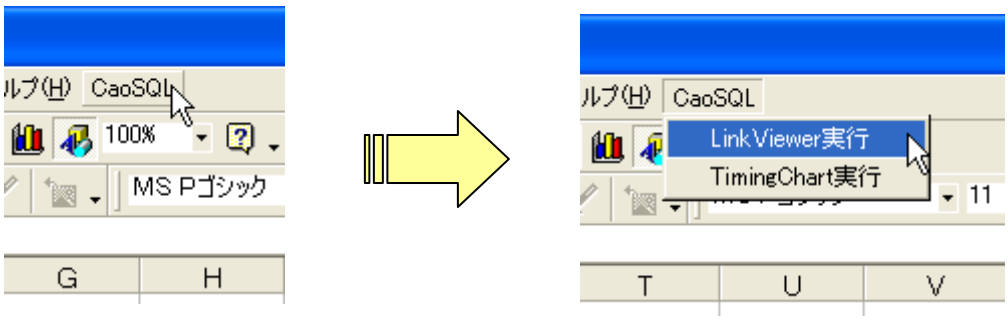


Figure 6-4 Add-in execution menu

Before executing Link viewer, you should check whether if the sheet name "CaoSQL-Connect" does not exist in the target file. If this sheet name already exists, it will be deleted and recreated.

Assuming that you are going to have several sheets of CaoSQL link status in one file. Once executing Link viewer and obtaining the link status sheet, you have to rename it before obtaining next link status sheet. Otherwise, it will be overwritten. You do not need to rename it if you are going to obtain the same link information as the original.

Figure6-5 shows the execution result of Link viewer.

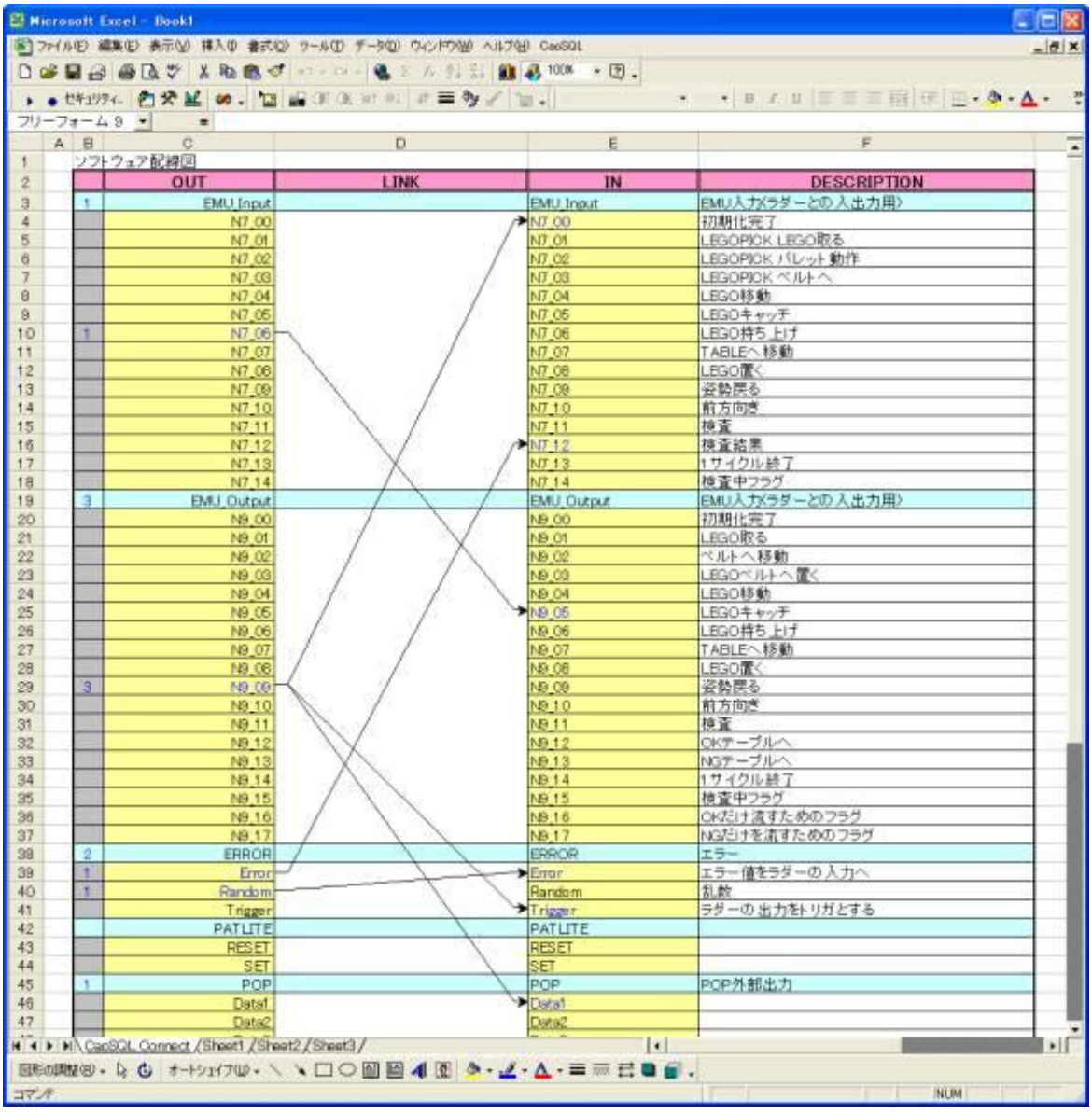


Figure6-5 Execution result of Link viewer

6.3. More details about Link viewer

6.3.1. Numbers of Link destination

This section describes the display of Link viewer. Figure 6-6 is a part of Figure6-5.

The left most column shows the number of links connected.

Lines with light blue represent controllers. The number of items linked is displayed in the leftmost cell.

In figure 6-6, you can see "2" in the leftmost cells of the "ERROR" line. This is because the two lines under "ERROR" line ("Error" and "Random") specify link destinations.

The right hand side of the table does not show any number of links.

An item name whose link destination is specified properly is written in blue ink.

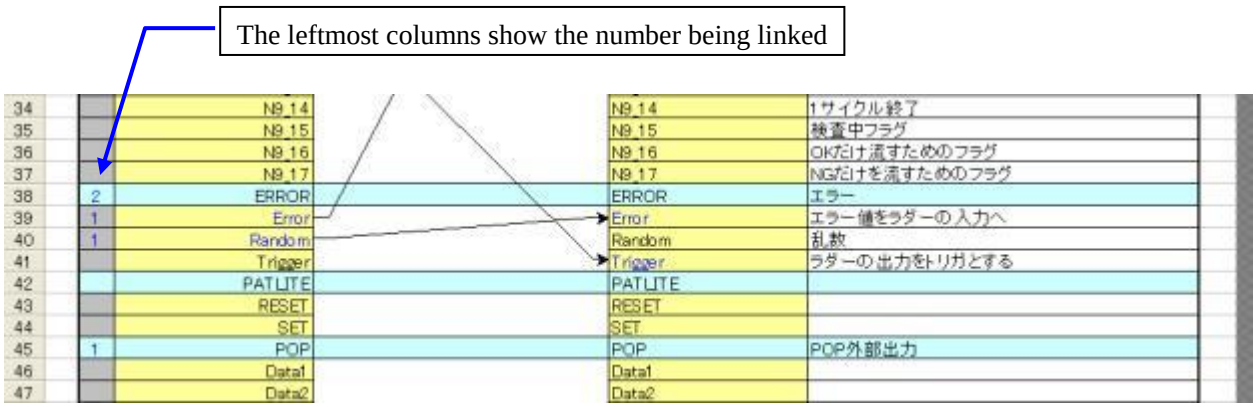


Figure 6-6 Number of link destination

6.3.2. Error display due to unknown link destination

If the link destination specified by the link viewer is invalid item, the link will be an error. In this case, an exclamation mark in red ink will be displayed in the leftmost cell.

If multiple link destinations are specified, an exclamation mark in red ink will be displayed in the leftmost cell even if only one invalid or unknown link destination exists. In this case, the error target item name is written in red ink as well.

Note that the link destination information cell (the leftmost cell) shows the total number of links that includes unknown or invalid links.

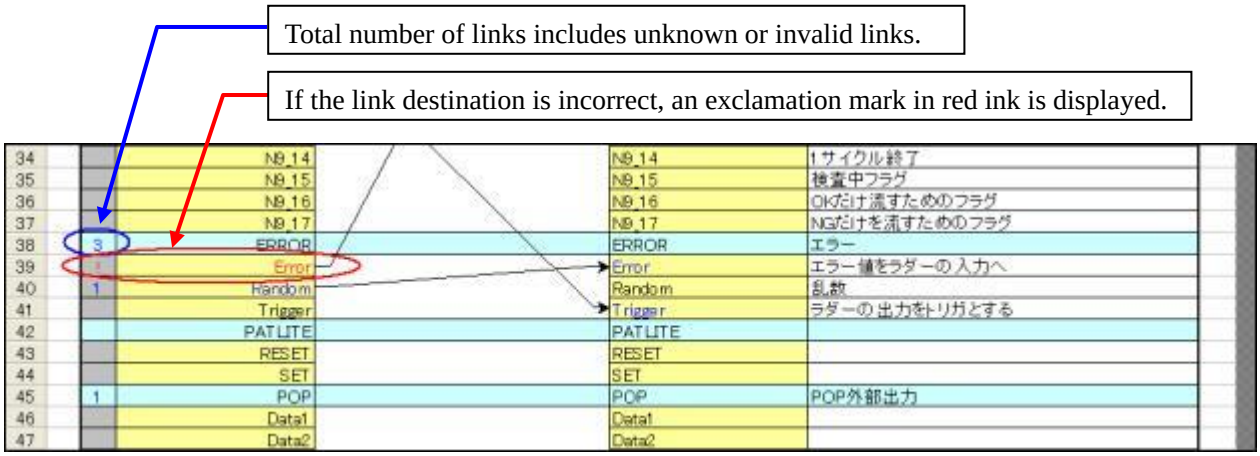


Figure6-7 Unknown link destination error

7. Timing chart

7.1. Overview

Timing chart[6] draws a chart based on the database recorded by CaoSQL.

Since this is Microsoft Excel add-in tool, the chart is displayed in Excel sheet.

You can choose desired CaoSQLItem and time period to display in the chart. You can choose a desired time period from 10, 20, or 30 seconds, starting from the time user specified.

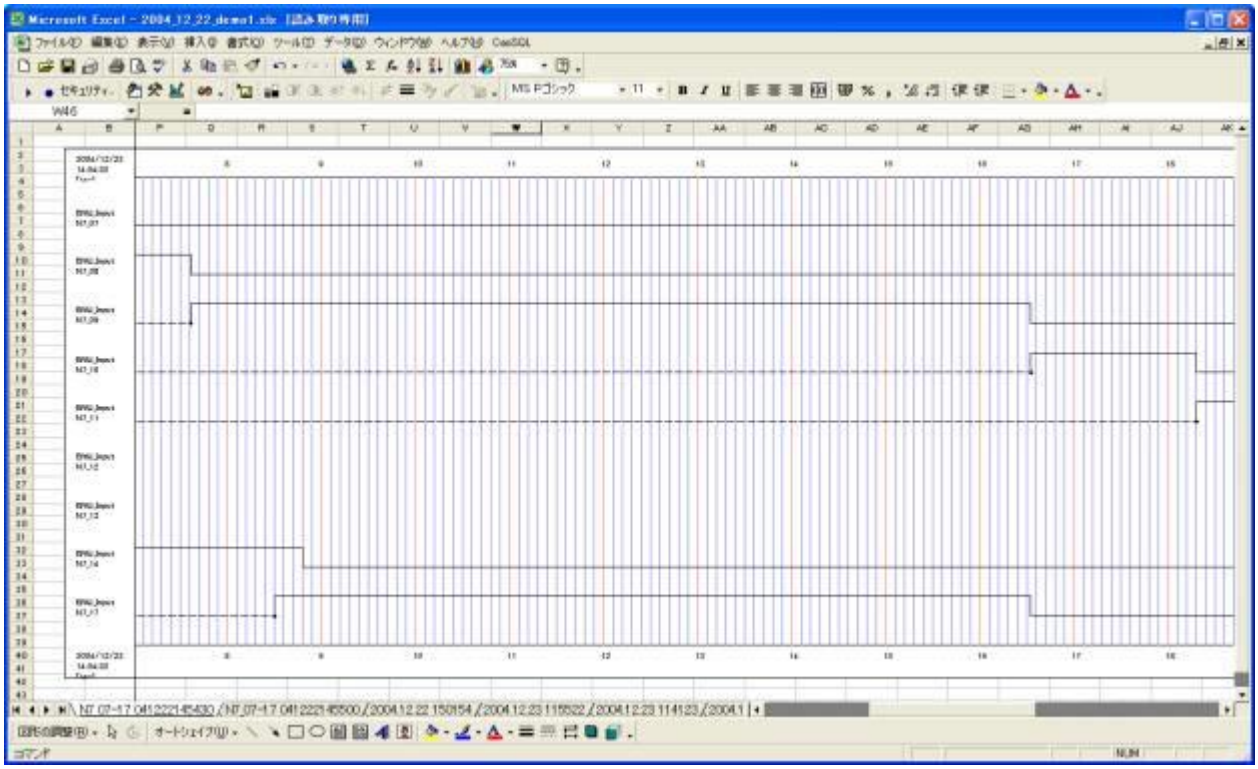


Figure7-1 Overview of Timing chart

[6] This tool uses a part of the result of the research that JSPMI (Japan Society for the Promotion of Machine Industry) executed with the assistance of the cycle race. This tool is open to the public to the support of the source though it is out of the support coverage.

7.2. Registering Add-in file

Before using the timing chart, you need to register Microsoft Excel add-in file.

For the procedure of add-in setup, refer to "6.2Registering Add-in".

7.3. How to use Timing chart

Once the add-in is registered, "Execute Timing Chart" is stored in the submenu of CaoSQL menu as shown in Figure 7-2. Click "Execute Timing Chart" command.

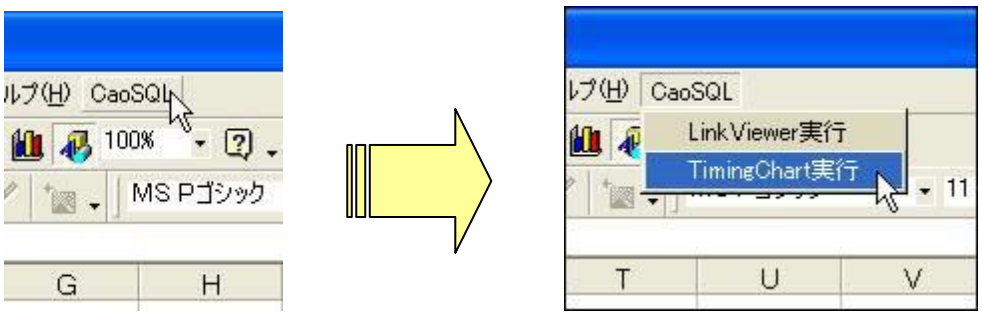


Figure 7-2 Menu after add-in is added

Once "Execute TimingChart" is clicked, the Timing chart setup wizard will be displayed. (Figure7-3)

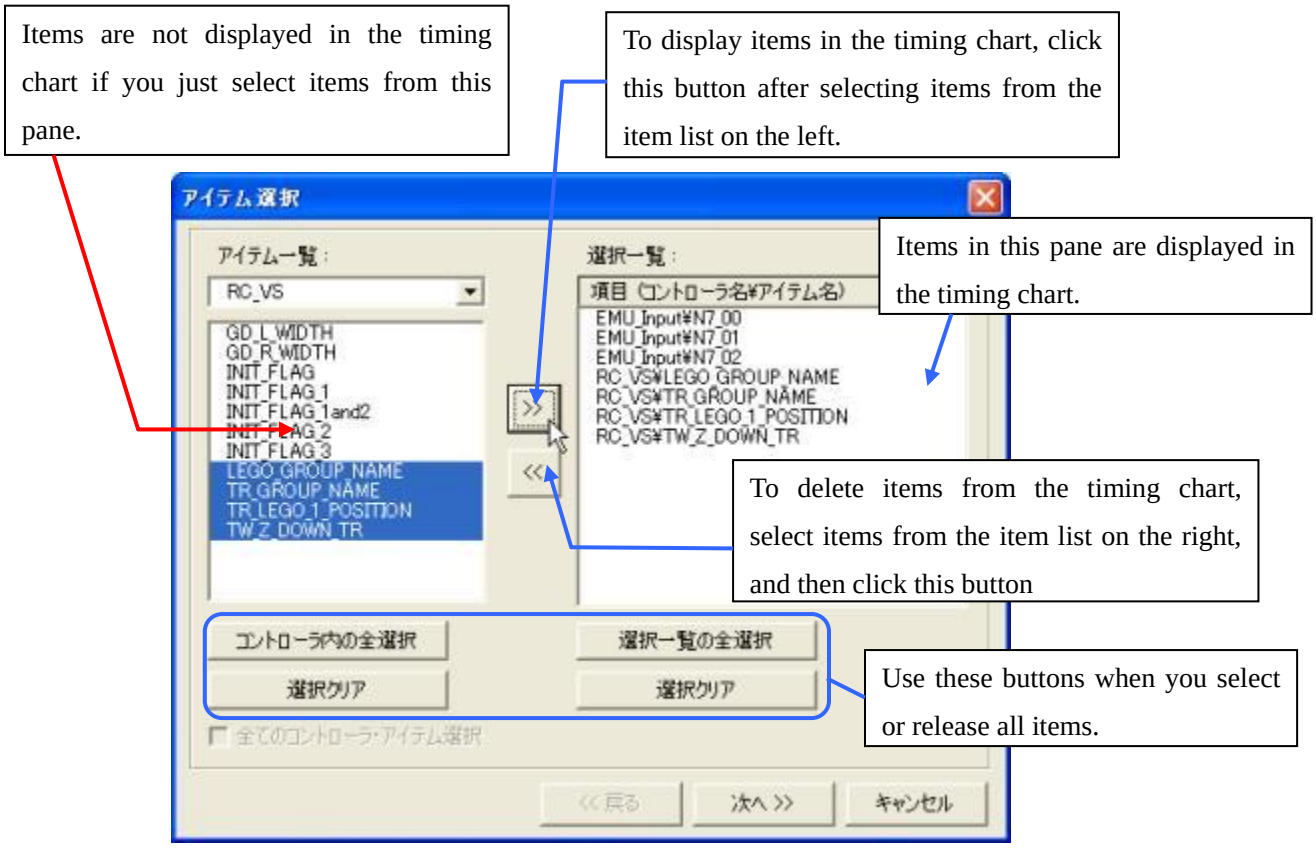


Figure7-3 Item selection of Timing chart setup wizard

From the item list on the left side, select items to display. You can select multiple items. Please note that items are not displayed in the timing chart unless the selected items are transferred to the item list on the right. To transfer the selected items to the item list on the right, click >> button at the center of this wizard. To release items from the timing chart, select items from the list on the right, and then click << button. The selection list lists up to 20 items.

Once all setup completes, click Next>> button.

The next page shows how to set the time period of the timing chart. (Figure7-4)

Figure7-4 Timing chart time period setup

Specify the start time and the duration of the timing chart. Duration of timing chart ranges from 10, 20, and 30 seconds. Invalid time setting will cause a warning message. Once all setup completes, click Next>> button.

Executing timing chart will create a new Excel sheet. Enter a sheet name on the text box as you like. The sheet name displayed in the default is the character string that consists of the time and data you have entered at the previous entry. For example, if you enter 2004/12/27 14:03:07 in the start time (see Figure7-4), the sheet name will be "2004.12.27 140307". If you give a name which is already exist in the file, default sheet name of Micosoft Excel will be used.

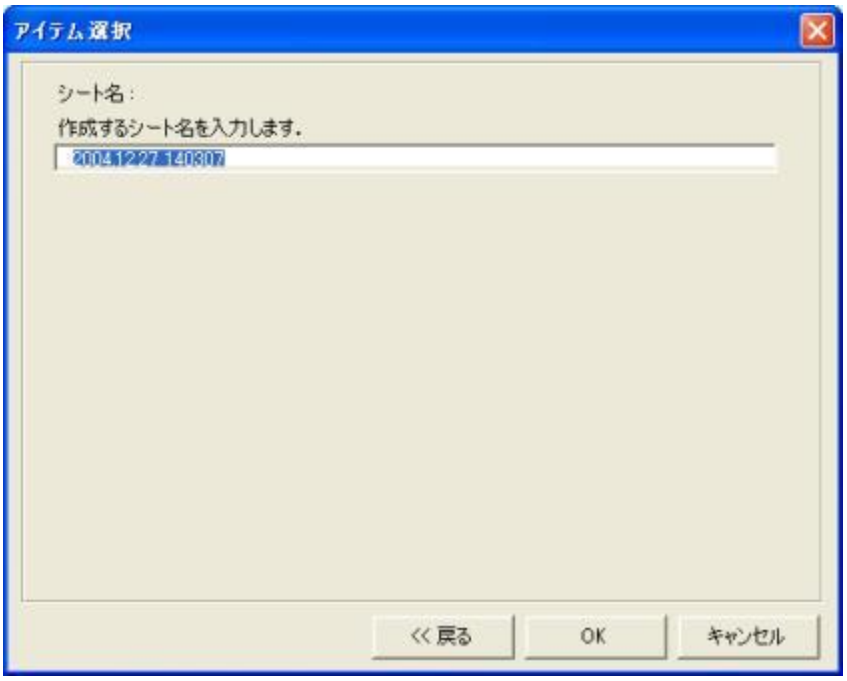


Figure7-5 Sheet name setting window

Once all setup completes, click OK. It may take longer to display the timing chart depending on the number of items.

7.4. More details about Timing chart

Timing chart utilizes CaoSQL History function and the chart is plotted based on the database recorded by CaoSQL. Basically, CaoSQL records an item value in the database when the value changes. Timing chart describes each item with on and off status. On-state represents item value other than logic zero, and off-state represents logic zero. Since this is a line chart, user can easily find the change timing of item value.

However, since there is no ways to determine the starting value is zero, other than zero, or empty, the beginning of each line is plotted with dotted line as unknown value.

The column of the item name is locked, so the item names and time always appears even if you scroll the window.

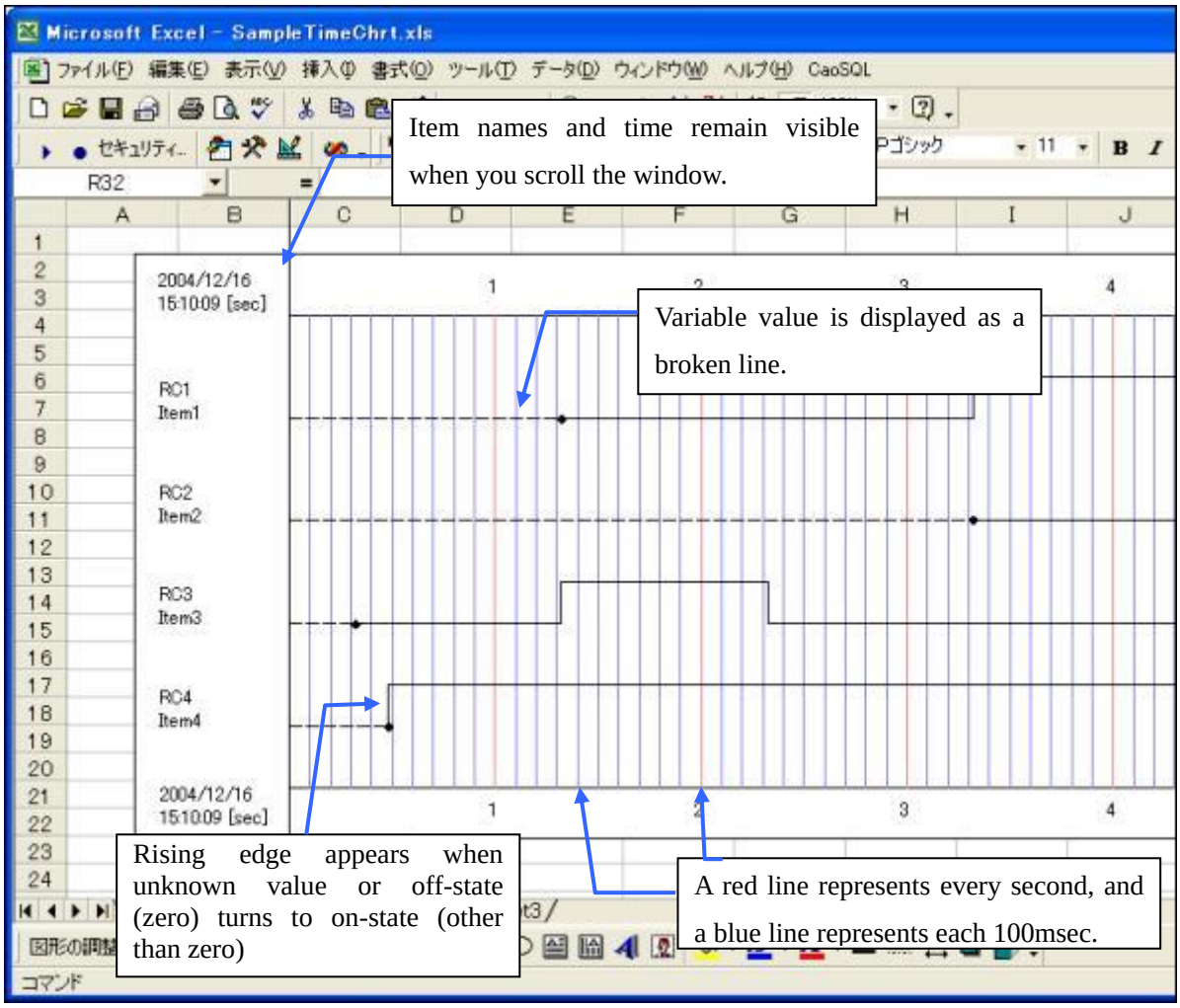


Figure 7-6 Window of timing chart

8. jCaoSQL

8.1. Outline

jCaoSQL[7] is a framework to control CaoSQL from java, and enables java application to access robot controller or other devices through CaoSQL interface. At present, all functions of CaoSQL can be used from java though some restrictions exist.

This chapter guides you to create CaoSQL client application of java by using jCaoSQL. Refer to "CaoSQL user's guide" for details in the CaoSQL interface.

8.2. Client development environment setup

The following shows how to develop a client by using three jCaoSQL files.

Table8-1 System files of jCaoSQL

File name	Outline
	Setting method
jCaoSQL.jar	java binary package of jCaoSQL
	Pass the file through ClassPath.
jCaoSQL.dll	Native code of jCaoSQL
	Put the file on the place where the client can refer. Or, register the place in %PATH% of the environment variable.
jcaosql_log.config	Log set up information of native code
	This is option. Put the file on the application execution folder when you need log information of the native code.

For more details about the way of client development, refer to the following descriptions and sample folder that is incorporated in the package

8.3. Sample programs

8.3.1. Creating and discarding an instance

CaoSQLEngine#createInstance, CaoSQLEngine#getController, and CaoSQLController#getItem are the Factory method pattern of CaoSQLEngine, CaoSQLController, and CaoSQLItem object, respectively. Each object creates corresponding COM instance. Therefore, if you delete these objects, you need to call "Release" explicitly b from the client application since java does not equip "destructor". You can refer the sample

[7] This tool is out of the support coverage.

programs with the following commands from sample folder.

```
> ant sample1
```

8.3.2. How to treat VARIANT type

In jCaoSQL, VARIANT class of ORiN maps to "java.lang.Object" class. This example shows how to determine the variable's VARIANT type of ORiN in the java client side.

With the instanceof operator, you can find which VARIANT type of ORiN corresponds to the variable of java. For instance, in this example, you will see the value corresponds to VT_I4 of ORiN's VARIANT.

```
if (value instanceof Integer) {
        // VT_I4 type variable
}
```

In order to distinguish whether the value is array type, obtain java.lang.Class from the object, and then call isArray method which is a Class object.

This sample is stored in the sample¥sample2 folder. You can refer the sample programs with the following commands from sample folder.

```
> ant sample2
```

8.3.3. Creating an Event sink

As the sample shows, jCaoSQL can receive events. However, CaoSQL does not issue events synchronously, so if you want to synchronize CaoSQL and java application from java side, you need to declare OnChangeItem method or OnChangeState method at the implementation of jp.co.denso.fa.CaoSQLControllerListner, or, declare *synchronized* in methods according to your needs.

Also, CaoSQLItem objects handed by arguments of OnChangeItem events must be discarded (call Release explicitly) just like 8.3.1.shows.

This sample is stored in sample¥sample3 folder. You can check the execution of sample with the following command from sample folder.

```
> ant sample3
```

8.3.4. Creating a custom class

To make programming and implementation easier, it might be practical to create a custom class, which is defined by user, by using CaoSQLjava. In this case, declare a custom class in jcaosql.properties, and then create a subclass of CaoSQLjava object.

The following shows the process.

Table8-2 Key list of jcaosql.properties

Key name	Description
caosql.engine.iid	Interface ID of CaoSQLEngine(no need to change this normally)
caosql.engine.progid	Program ID of CaoSQLEngine(no need to change this normally)
caosql.engine.class	CaoSQLEngin class created by factory Default "jp.co.denso.fa.jcaosql.CaoSQLEngine"
caosql.controller.class	CaoSQLController class where factory tries to create Default "jp.co.denso.fa.jcaosql.CaoSQLController"
caosql.item.class	CaoSQLItem class where factory tries to create Default "jp.co.denso.fa.jcaosql.CaoSQLItem"

For example, to use com.example.MyCaoSQLItem class whose super class is CaoSQLItem, create a new file with the name of "jcaosql.properties", and then map the value of <com.example.MyCaoSQL> into the key < jcaosql.item.class >, and then allocate this properties file in the execution root. With this process, factory class will create com.example.MyCaoSQLItem class when CaoSQLItem is created.

This sample is stored in the sample¥sample4 folder. You can refer the sample programs with the following commands from sample folder

```
> ant sample4
```

8.3.5. Dynamical adding and deleting items

In jCaoSQL, just like CaoSQL, users can add items dynamically. Note that user can delete only dynamically added items.

This sample is stored in the sample¥sample5 folder. You can refer the sample programs with the following commands from sample folder

```
> ant sample5
```