

CaoSQL

User's guide

Version 1.9.2

October 2, 2018

[Remarks]

[Revision history]

Date	Version	Content
2006-03-09	1.0	First edition.
2006-08-20	1.1.0	Supported the first dimension array data in History function. Added the data type conversion function at data-writing.
2006-09-22	1.1.1	Added Data comparison in array function (Compare Arrays option)
2007-12-26	1.2.0	Added Array element distribution link (Extract Linking option)
2008-01-30	1.2.1	Added Provider setting tips.
2008-12-12	1.3.0	Added ShellExecute into trigger action.
2009-05-22	1.3.1	Added commands into Execute of Controller class (\$CSQ_RESET_SCANTIMES\$).
2009-05-26	1.3.2	Corrected CaoSQLController::SettingData, and CaoSQLItem::SettingData
2009-06-10	1.3.3	Added <i>Decomposition of VT_I4 type into two VT_I2 type data in Tips for setting.</i>
2009-07-13	1.4.0	Added file class variables. Added an inactivation option for each item
2009-10-12	1.5.0	Added unary operation into Binary operation function (Ex. NOT, ABS)
2009-11-19	1.5.1	Added array type item and the alias function.
2010-02-05	1.5.2	Added items grouping function
2010-02-05	1.6.0	Added Link destination property selection function.
2011-03-19	1.6.1	Added commands to Execute of Controller class (\$CSQ_SCAN_ONETIMES\$).
2011-03-28	1.6.2	Added Item ID specification in the RAC server function.
2011-06-20	1.7.0	Added Script execution function.
2011-07-03	1.7.1	Corrected the Registry structure information
2011-09-05	1.7.2	Added the property acquisition command of RAC server function
2011-10-21	1.7.3	Improved the speed of RAC server function.
2013-02-20	1.7.3	Modified the DENSO Robot mode (supported RC8).
2013-12-14	1.8.0	Improved the Binary operation function. (added the second operation)
2014-01-13	1.8.1	Improved the speed of History function. (added "No Conversion To BSTR" option)
2015-05-21	1.8.2	Added CoAP server function.
2015-07-06	1.8.3	Added "Saving Binary data into a file" of History function.
2015-09-01	1.8.4	Added Parameter copy menu in History function.
2015-11-06	1.8.5	Added Database information into system log of History function. Added option string entry for parent object at Item creation.
2016-11-22	1.8.6	Modified the bug that cannot list up all the items located below the group in which the controller has no items.
2016-12-19	1.9.2	Modified the description of Watchdog timer.
2018-10-02	1.9.2	Added operation confirmed databases.

[Dependence module]

No.	Dependence module
1	MDAC (Microsoft Data Access Components) 2.7
2	Internet Explorer 5.01 SP2 or later

Contents

1. Introduction.....	7
2. Overview of CaoSQL.....	8
2.1. Overview.....	8
2.2. Object model.....	9
2.2.1. Type of CaoSQLController.....	9
2.2.2. Types of CaoSQLItem.....	10
2.3. Operation model.....	10
2.4. Sample program.....	12
2.4.1. Visual Basic.....	12
2.4.2. SQL.....	12
2.4.3. Active Server Page (ASP).....	15
2.5. Error processing.....	17
2.5.1. Processing for CaoSQLEngine error occurrence.....	17
2.5.2. Processing for CaoSQLController error occurrence.....	17
2.5.3. Processing for CaoSQLItem error occurrence.....	17
2.5.4. Processing for CaoSQLHistory error occurrence.....	18
3. Function of CaoSQL.....	19
3.1. Function list.....	19
3.2. Event notification function.....	21
3.2.1. OnChangeltem event.....	22
3.2.2. OnChangeState event.....	22
3.3. Trigger function.....	22
3.4. DDE server function.....	23
3.5. Network DDE server function.....	25
3.6. Log function (Error logging).....	28
3.7. Cascade connection function of CaoSQL.....	29
3.7.1. CaoSQL provider.....	30
3.7.2. RAC provider.....	32
3.7.3. e-CAP provider.....	32
3.7.4. CoAP provider.....	32
3.8. History function (Data logging).....	33
3.8.1. System information recording.....	35
3.8.2. Converting character string of an item value.....	36

3.8.3. Saving binary data into a file	36
3.9. Secondary DB switch function.....	36
3.10. Array element extraction function.....	37
3.11. Data masking function	37
3.12. BCD conversion function	37
3.13. Binary operation function	38
3.13.1. The second operation function	39
3.14. Data type conversion function	40
3.15. Dead band setting function	40
3.16. Valid range setting function	40
3.17. Item link function	41
3.17.1. Selecting Link destination property	42
3.18. Inactivation function	42
3.19. Array type item function	42
3.20. Alias function	43
3.21. RAC Server function.....	43
3.21.1. GET command	44
3.21.2. PUT command.....	45
3.22. CoAP server function.....	45
3.22.1. GET command	46
3.22.2. PUT command.....	46
3.23. Item dynamic registration/deletion function.....	47
3.24. Script execution function.....	50
3.25. CSQ command	52
3.26. The list of available data type information.....	53
4. CaoSQLConfig	54
4.1. Outline	54
4.2. Operation method	54
4.2.1. Tab input	54
4.2.2. Menu	68
4.3. DENSO Robot mode.....	77
4.3.1. Add controller (DENSO Robot mode)	77
4.3.2. Add item (DENSO Robot mode).....	80
4.3.3. Loading configuration file for normal mode CaoSQL	83
4.4. Registry structure information	85
4.5. Tips for Setting	86

4.5.1. Decomposition of array element.....	86
4.5.2. Decomposition of bit element.....	87
4.5.3. Polynomial operation.....	88
4.5.4. Conditional Item Link.....	89
4.5.5. Decomposition of one VT_I4 type data into two VT_I2 type data.....	90
4.6. Provider setting Tips.....	92
4.6.1. DataStore provider.....	92
4.6.2. DataBase provider.....	94
5. CaoSQL tutorial.....	97
5.1. Outline.....	97
5.2. Database setting by Access2000.....	97
5.3. Controller and Item registration by CaoSQLConfig.....	98
5.4. Creating a client application.....	98
5.5. Execution of sample.....	101
Appendix A.....	103
Appendix A.1. CaoSQL API list.....	103
Appendix A.2. CaoSQLController state transition chart.....	118
Appendix A.3. CaoSQLController state transition chart.....	119

1. Introduction

This document is a user's guide of CaoSQL that works as a client application of CAO (Controller Access Object).

CaoSQL is a middleware for data management that collects data from multiple FA equipment and then provides these data to CaoSQL client application, such as operation management system and work instruction software.

Although the main purpose of CaoSQL is to provide data collected from various FA equipment to the upper application, it also registers collected data to relational database. The registered data can be analyzed by using general-purpose front end software, such as Microsoft Access. This feature is especially practical when you collect specific data, such as “Number of production every day from October 1, 2002 to October 31, 2002, the average, variance, and standard deviation”.

CaoSQL can handle various commercial database engine such as Access, SQL-Server, and ORACLE, therefore, you can adapt CaoSQL to existing corporate system easily¹. Also, the system structure of CaoSQL is common for both in distributed computing environment and in environment that uses only one computer. In the distributed computing environment, application can collect provider's data through cascaded CaoSQLs, this realizes large number of FA equipment connection without degrading performance even in a large scale of facilities, such as factory..

As mentioned above, CaoSQL (and the combination of CaoSQL and CAO) serves as a system that provides a solution for a subject involving variety of equipment and software, so using CaoSQL makes it easy to develop systems.

This document explains the specification of CaoSQL.

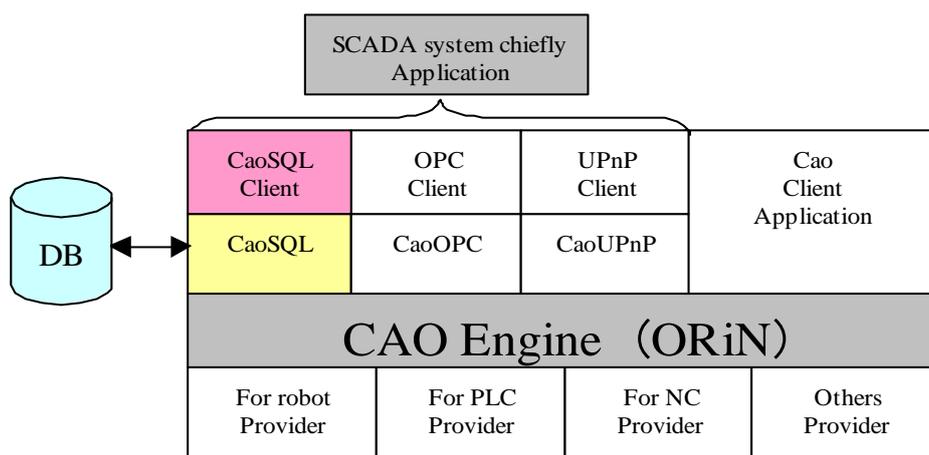


Figure 1-1 Form of CAO application

¹ There are some differences in the performance and SQL language specification by the database engine.

2. Overview of CaoSQL

2.1. Overview

The figure below shows the structure of CaoSQL. CaoSQL is a client tool of CAO. CaoSQL creates and manages CAO objects, such as CaoController, CaoVariable, and CaoFiles, based on the information registered in the csq file², which is a system file created by CaoSQLConfig.exe. Since CaoSQL client obtains such data through CaoSQL interface, you do not need of considering CAO interface. CaoSQL is a middleware that is specially designed for data reading and writing, so the structure is relatively simple comparing to CAO interface specification (see Object model). Also, it equips unique functions that CAO engine does not have (see Function of CaoSQL).

CaoSQLConfig Tool is used to configure parameters necessary to CAO objects, such as CaoController, CaoVariable, and to configure motion options of CaoSQL. CaoSQLTester allows you to execute CaoSQL methods and to check the retrieval and setting of property. CaoSQLCmd tool is a console application that executes CaoSQL method from command line and obtains and configures certain properties. CaoSQLLauncher starts and stops CaoSQL. For CaoSQLConfig, refer to ⁴CaoSQLConfig. For other tools, refer to "[CaoSQLTools user's guide](#)".

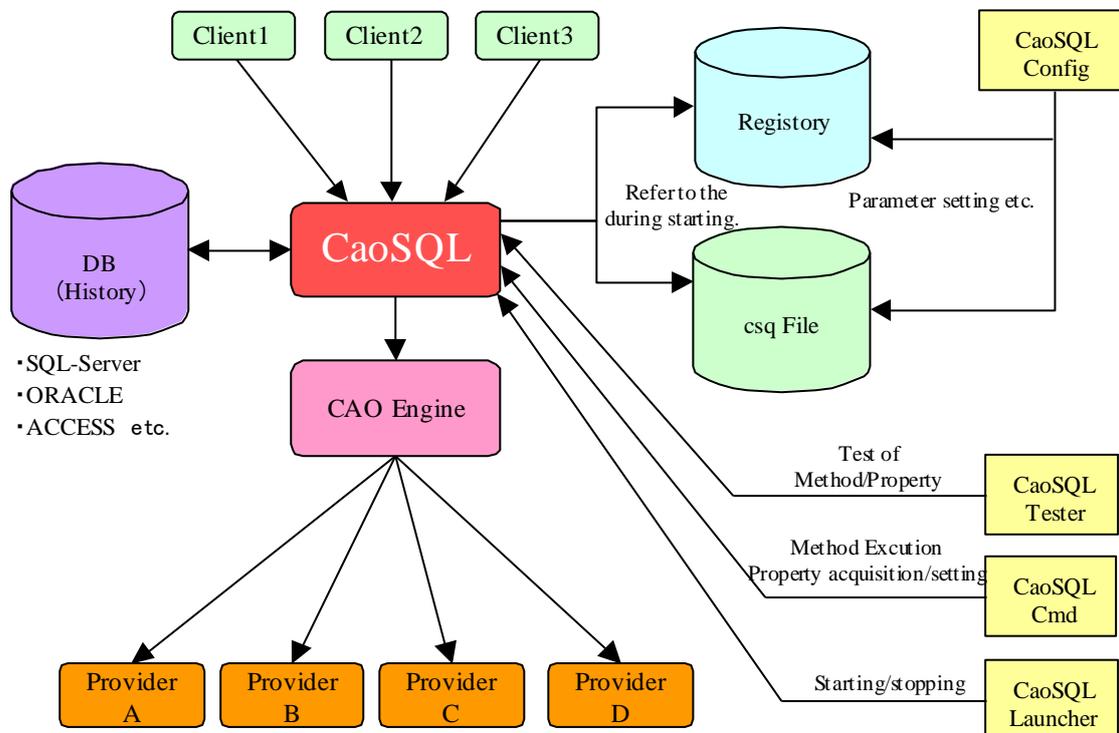


Figure 2-1 CaoSQL outline chart

² It is configuration file made in CaoSQLConfig.

2.2. Object model

CaoSQL has a simple object model as follows. Only CaoSQLEngine can be generated from the outside³. CaoSQLController and CaoSQLHistory objects are operated through CaoSQLEngine interface.

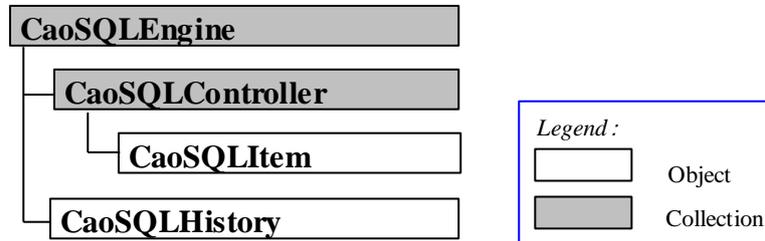


Figure 2-2 Structure of CaoSQL class

Table 2-1 Functions offered by CaoSQL class

Class name	Function offered
CaoSQLEngine	Collection class of CaoSQLController. This maintains the entire setting of CaoSQL. This is an only object that can be directly generated from the outside. The subordinate objects are obtained and created through this interface.
CaoSQLController	Collection class of CaoSQLItem. This corresponds to CaoController object of CAO and maintains its setting. The sampling thread, which executes polling processing and event processing, is created by the unit.
CaoSQLItem	This class corresponds to CaoVariable of CAO and maintains its setting. Use this object when you read or write values into items configured by CaoSQLConfig.
CaoSQLHistory	This class maintains the settings of database used in CaoSQL History function. You can execute SQL sentence directly to database.

2.2.1. Type of CaoSQLController

There are two types of CaoSQLController; normal type and alias type. As the object model above shows, normal CaoSQLController is processed just like CAO's CaoController. Alias controller is a controller that comprises alias items. You cannot specify providers and computers for the alias controller.

³ Please refer to appendix 5.5.Appendix A.1 or [CaoSQLObjects](#) for the list of the function of each class.

2.2.2. Types of CaoSQLItem

Unlike CaoSQLController, there are three types of CaoSQLItem; general, arias, and array type. General type's CaoSQLItem is processed just like CAO's CaoVariable.

For an arias type's CaoSQLItem, you can register a general and/or array type's CaoSQLItems as a different item with a different name so that users can link values to them. Array type's CaoSQLItem incorporates multiple item's values as an array so that users can read/write values from/in items.

2.3. Operation model

In CaoSQL processing, each CaoSQLController has one CaoController to execute a thread (sampling thread).

Each running thread periodically obtains CaoVariable's value that is stored in CaoSQLItem object under CaoSQLController object, with specified sampling interval, and then caches the value.

CaoSQL client refers the internal cache by using a reading function. If the periodically obtained value differs from the cached value, CaoSQL client rewrites the value of cache, and, if necessary, writes it in History, issues an event, such as OnChangeItem event, OnChangeState event, or executes other processing. Note that only the following variable types can monitor the value change.

Table 2-2 VT type that can observe value change

Type of VARIANT	Description	Type of C language
VT_EMPTY	(Empty)	-
VT_NULL	NULL	-
VT_I1	One byte integer type signed	char
VT_UI1	One byte integer type unsigned	unsigned char
VT_I2	Two byte integer type signed	short
VT_UI2	Two byte integer type unsigned	unsigned short
VT_I4	Four byte integer type signed	long
VT_UI4	Four byte integer type unsigned	unsigned long
VT_R4	Floating point of single precision type	float
VT_R8	Floating point of double precision type	double
VT_CY	Currency type	-
VT_DATE	Date type	DATE(double)
VT_BSTR	(UNICODE) Character string type	BSTR
VT_ERROR	Error type	-
VT_DISPATCH	IDispatch pointer	IDispatch*
VT_UNKNOWN	IUnknown pointer	IUnknown*

For other type's variables, although values of caches are updated at any time, the processing accompanied by the value change, History writing, link, OnChangeItem event issuance and other processing are not executed. The acquisition time (DateTime property) is also updated in the cache updated timing.

However, if CaoSQL fails to obtain values from CAO, an error is set to item's status. In this case, neither the cache value nor the acquisition time are updated⁴.

On the other hand, to write a value into a provider, CaoSQL requests CAO to write the value into the provider once a client executes a writing function.

Since these processing are done by each thread, it hardly influences on other controller's performance even if a certain controller's response is slow. In addition, the priority of this thread can be explicitly specified by using CaoSQLConfig tool. (For details, refer to "CaoSQLTools user's guide".)

For the list of the controller's thread and item's status, refer to CaoSQLObjects.

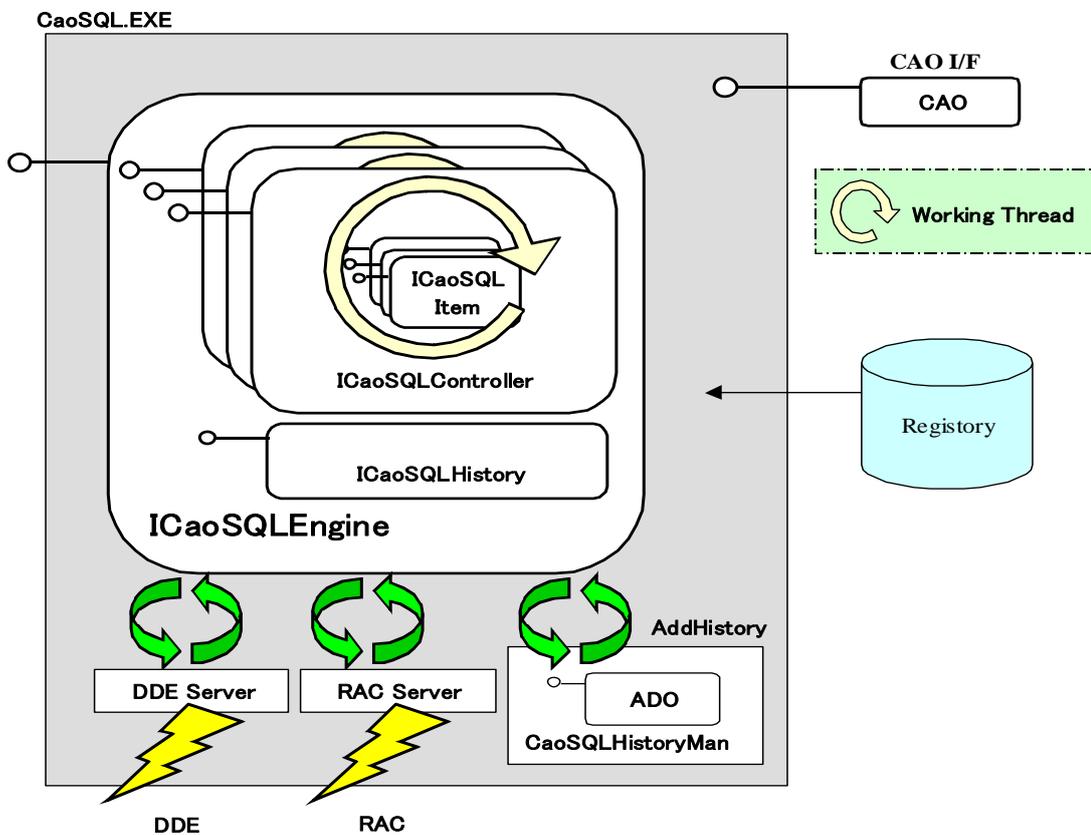


Figure 2-3 CaoSQL working model

⁴ If the acquisition of all items in Controller object fails, a penalty that delays one cycle to give priority to other controller's processing is automatically charged.

If the even distribution option (distribute equally) is enabled, the time delay (sampling interval / Item count) is inserted every time an item value is obtained. If Sampling off (Startup off) is set, the cache value is updated whenever the data reading.

2.4. Sample program

2.4.1. Visual Basic

The following shows sample programs of CaoSQL. Programming language used here is Microsoft Visual Basic 6.0. You can implement it by using other programming language as well.

First of all, write the following program to create a CaoSQLEngine instance.

```
Dim csqEng As Object
Set csqEng = CreateObject("CaoSQL.CaoSQLEngine")
```

"CaoSQL.CaoSQLEngine" is program ID of CaoSQL. If CaoSQL has already registered in Reference page, you can create an object with *New* operator as follows.

```
Dim csqEng As New CaoSQLEngine
```

Next, insert the value of item *I1* of *RC1* controller into a variable 'x'.

```
Dim x As Variant
x = csqEng.Controller("RC1").Item("I1").Value
```

Here, because Controller property, Item property, and Value property are the default property of each class, this program can also be written as follows.

```
x = csqEng("RC1")("I1").Value
```

On the other hand, to set the value of the same item to "10", write as follows.

```
csqEng("RC1")("I1").Value = 10
```

Since the above sample programs are written in easy-to-understand way, these are not practical for the actual use because CaoSQLController class and CaoSQLItem class instances are created and referred every time and then deleted whenever the reference finishes. In actual application, object interface should be saved and reused unless the object interface is used only one time. See the following program.

```
Dim x As Variant
Dim csqCtrl As CaoSQLCmd
Dim csqItem As CaoSQLItem
Option Explicit

Set csqCtrl = csqEng.Controller("RC1")
Set csqItem = csqCtrl.Item("I1")
Debug.Print "I1 = " & csqItem.Value
```

2.4.2. SQL

CaoSQL can detect the value change of items and record the change in database. This function allows users to analyze various data by using SQL language. The following shows how to execute SQL with CaoSQLHistory interface.

CaoSQLHistory interface equips a property that obtains contents specified by the Database setup dialog of

CaoSQLConfig tool. It also can obtain History table name, and in addition to that, it has Execute method that executes SQL sentence and returns the recordset of the execution result. With this Execution method, you can analyze History table without executing Open or Close of database.

For example, to extract the recordset of Item *I1* in Controller *RC1*, write a program as follows.

```
Dim csqHis As CaoSQLHistory
Dim rs As ADODB.Recordset

Set csqHis = csqEng.CaoSQLHistory
Set rs = csqHis.Execute("SELECT * FROM caosql_history _
WHERE controller_name = 'RC1' AND item_name = 'I1' ")
```

Note that the value of the underlined *caosql_history* must be the same table as that is configured by CaoSQLConfig tool. To achieve this operation in program, use TableName property as follows. In this case, be sure to enter blanks before and after the TableName.

```
Set rs = csqHis.Execute( "SELECT * FROM " & csqHis.TableName & _
" WHERE controller_name = 'RC1' AND item_name = 'I1' ")
```

Other SQL samples are shown as follows.

Table 2-3 Inquiry example to History table

Inquiry example	SQL sentence (Underlined parts change arbitrarily.)
Number of records of each controllers	SELECT controller_name, COUNT(*) FROM caosql_history GROUP BY controller_name
Number of records of each item in a certain controller	SELECT controller_name, item_name, COUNT(*) FROM caosql_history WHERE controller_name = 'RC1' GROUP BY controller_name, <u>item_name</u>
Recordset of a certain item	SELECT * FROM caosql_history WHERE controller_name = ' <u>RC1</u> ' AND item_name = ' <u>I1</u> '
Recordset of the latest data (Flags is 1).	SELECT * FROM caosql_history WHERE flags = 1
The maximum and minimum values of each item in a certain controller	SELECT controller_name, item_name, MIN(item_value), MAX(item_value) FROM caosql_history WHERE controller_name = ' <u>RC1</u> ' GROUP BY controller_name, <u>item_name</u>
Total, average, and standard deviation of each item in a certain controller	SELECT controller_name, item_name, SUM(CAST(item_value AS decimal)), AVG(CAST(item_value AS decimal)), STDEV(CAST(item_value AS decimal)) FROM caosql_history WHERE controller_name = ' <u>RC1</u> ' GROUP BY controller_name, <u>item_name</u>
Recordset in a certain specified time	SELECT * FROM caosql_history WHERE item_time BETWEEN '2002/03/18 15:09:35' AND '2002/03/18 15:09:40'

Inquiry example	SQL sentence (Underlined parts change arbitrarily.)
Recordset of an item that is larger than average of all item value of a certain controller ⁵	SELECT * FROM caosql_history WHERE item_value > (SELECT AVG(CAST(item_value AS decimal)) FROM caosql_history WHERE controller_name = 'RC1')
Delete a certain controller's all record	DELETE FROM caosql_history WHERE controller_name = 'RC1'

⁵ Please note that it is not possible to use Sum function in WHERE phrase.

2.4.3. Active Server Page (ASP)

CaoSQL, which is installed by ORiN2 SDK, contains ASP sample programs in the following folder.

Sample program directory: "< ORiN2 installation directory folder > \CaoSQL\Sample\ASP"

The sample program is composed of the following three files.

(1) A.asp

- Display each Control specified by CaoSQLConfig in the server, and then acquire items

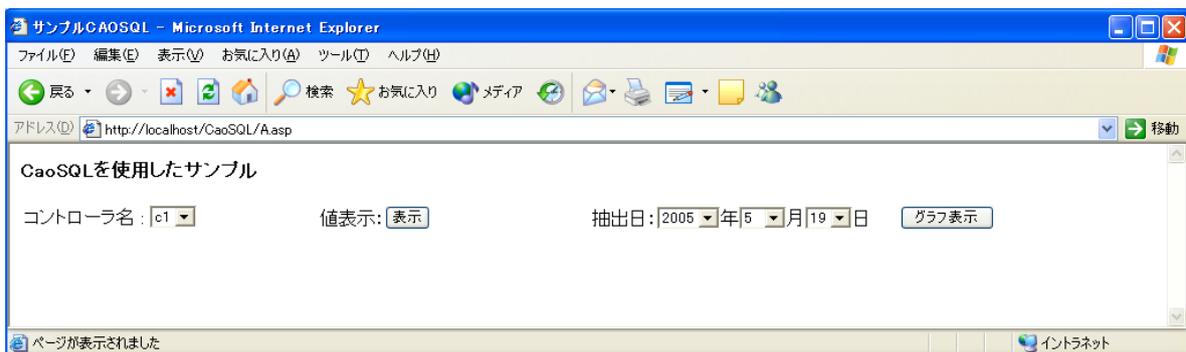


Figure 2-4 A.asp

(2) B.asp

- Acquire data from DB by using History object of CaoSQL.
- Pass the SQL sentence that extracts conditions into Execute of History object. Acquire the record.
- Based on the acquired record, plot a bar chart by using OfficeXPweb component.

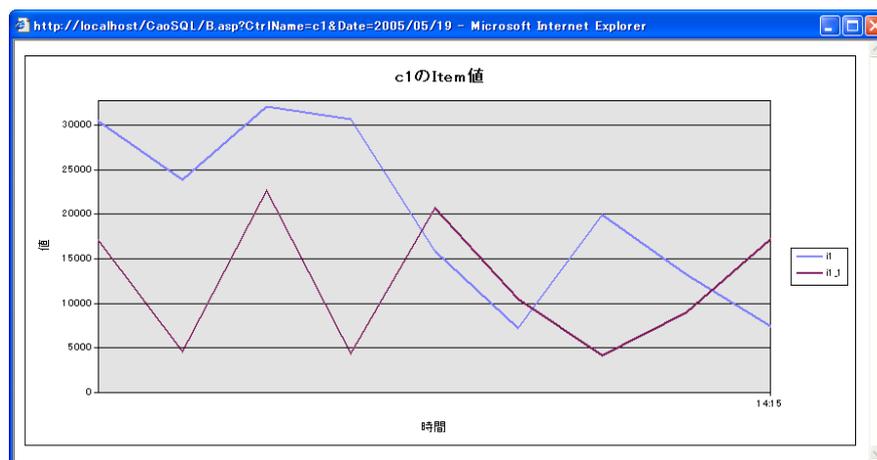


Figure 2-5 B.asp

(3) C.asp

- C.asp is invoked by A.asp. This outputs a value of the specified controller's item.

- Since this is not updated automatically, click Renew button to update.

コントローラ名	値
i1	1495
i1_1	28975

Figure 2-6 C.asp

2.4.3.1. Installation procedure

- (1) Install a component
 - Install ORiN version 2.
 - Install IIS (Internet Information Service).
 - Install OfficeXPweb components.

Table 2-4 Source of Office XP web component

File name	Source URL
owc10.exe	http://www.microsoft.com/downloads/details.aspx?FamilyID=982b0359-0a86-4fb2-a7ee-5f3a499515dd

- (2) Create a virtual directory with IIS.
 - Create a virtual directory with the management tool of IIS.
 - Set the reference destination of the virtual directory as follows.
 - <ORiN2 install directory>\CaoSQL\Sample\ASL
- (3) Security setting of CaoSQL
 - This is option. To start this setting as a service, execute "CaoSQL.exe/service" from the command prompt.
 - Start DCOM setting tool "dcomcnfg.exe". Open the Property dialog of CaoSQL.
 - Configure the security setting along with the security policy of the virtual directory created. (access control and start access control, etc.)
- (4) Confirm the operation

Open "Http://localhost/< virtual directory name >/ A.asp" by IE a browser.

Confirm the setting of DCOM again if the sample does not run properly.

2.5. Error processing

This section explains what process is executed when an IF error of CAO occurs in CaoSQL, and how the client side recognizes the error.

2.5.1. Processing for CaoSQLEngine error occurrence

Expected errors occurred in CaoSQLEngine are, for example, the CaoEngine acquisition failure, and CaoSQLConfig setup information reading failure. Since these errors are related to the CaoSQL fundamental operation, once this error occurs, the system outputs an error log and CaoSQL.exe stops its operation.

2.5.2. Processing for CaoSQLController error occurrence

[Error at the initialization]

This is an initialization error at the reading failure of CSQ file or other reasons. In this case, CaoSQLController cannot be recovered. An error is output to the log, and CaoSQLController is not generated.

[Error during startup]

An error at the CaoController generation and at the generation of the thread applies to this error. In this case, the thread is not executed, although CaoSQLController's instance is created. The error is output to the log, and the status of CaoSQLController is set to Deactive.

2.5.3. Processing for CaoSQLItem error occurrence

[Error at the initialization]

This is an initialization error at the reading failure of CSQ file or other reasons. CaoSQLItem object is not generated. The error is output to the log.

[Error during startup]

This applies to an error that is generated at CaoVariable obtainment. In this case, the error is output to the log, and the state is saved in Status of CaoSQLItem.

[Error during operation]

Basically, process is retried when an error occurs, so the error does not affect sampling threads. It means, the sampling thread keeps running until CaoSQLItem releases the control object by stopping the thread or terminating the process (output an error to the log). Also, an Item records own state in Status whenever sampling is performed. You can obtain the state by using State property from CaoSQLItem

interface.

Table 2-5 State of CaoSQLItem

State	Details
csItemSucceeded	The value of the item was normally acquired/set.
csItemFailed	Failed to acquire/set the item's value.
csItemUndefined	The value of the item is irregular.

2.5.4. Processing for CaoSQLHistory error occurrence

[Error at the initialization]

This error applies to an initialization error, such as CSQ file reading failure or other reasons. The error is output to the log.

[Error during startup]

This error is mainly applies to the database connection failure. In this case, an error is output to the log, and then the system tries to connect the secondary database if Secondary DB switch function is enabled.

[Error at the operation]

This applies to the failure of record addition to a table. In this case, an error is output to the log, and then the system tries to connect the secondary database if Secondary DB switch function is enabled.

3. Function of CaoSQL

3.1. Function list

CaoSQL has the following functions. These functions support to establish data links with multiple FA devices with minimum programming.

Table 3-1 CaoSQL function list

Function	Description
[Functions applied to the entire process]	
Log (Error logging)	This function outputs the log that displays entire operation status, such as, creation and deleting of objects, start and stop of CaoSQL, error occurrences. Users can select the log output destination from the followings: event viewer, debug viewer, or console.
History function (Data logging)	History function records an item's value in the database when the item's value changes or Snapshot method is explicitly called. (Refer to 'Secondary DB switch function')
Secondary DB switch function	If this function is enabled, the system automatically switches the data recording destination to the secondary database when the database server is in error-state while the item's value is recorded into the database with History function. With this function, data is registered without fail.
DDE server function	This is a DDE server function. This function allows to obtain item's value from DDE client application, such as Microsoft Excel, without programming.
RAC server function	This is RAC server function. This function allows to exchange data with CaoSQL through network with RAC commands by using RAC provider.
CoAP server function	This is CoAP server function. This function allows to exchange data with CaoSQL through network by CoAP by using CoAP provider.
[Functions running by the controller]	
Cascade connection	While this function is used under the distributed computing environment, if CaoSQL provider is used, information collected by respective computer's CaoSQL can be integrated into one computer. (Remarks) Similar function can be achieved by using RAC provider or others.
Sampling function	Performing data sampling and caching by each controller asynchronously will realizes the prompt response to the value request from the client application.

Watch-dog function	If this function is enabled, specified processing is executed when the specified sampling interval delays.
Trigger function	This function executes specified processing for a sampling thread if a certain conditions are met.
Automatic connection retrieval function	This function recreates related CAO objects if Controller's status remains error. Use this function to recover the connection status automatically when device shuts down during connection.
Item dynamical adding function	If this function is enabled, items can be dynamically recorded or deleted after CaoSQL startup. You can delete items that are dynamically added only.
Script execution function	This function executes VB script. Read VB script configured by CaoSQLConfig at the startup, and execute subroutine Main in the script at the timing of sampling.
[Functions running by the item]	
Array element extraction function	When the value of an item is an array, only a necessary element can be extracted. Delimit the index number of the element to be extracted by the comma, and enclose it with brackets to add it to the item name. Data processing order : (at the reading) 1.
Masking	This masks an item's value. Set the mask if you need to extract specified bits only. This function is available only for one byte, two bytes, or four bytes signed/unsigned integer type. Data processing order : (at the reading) 2. (at the writing) 3
BCD conversion	This converts an item's value to BCD (Binary Code Decimal) at the time of reading and writing. This function is available only for one byte, two bytes, four bytes signed/unsigned integer. Data processing order : (at the reading) 3. (at the writing) 2
Binary operation function	This performs binary operation. Available operators are ; addition (+), subtraction (-), multiplication (*), division (/), modulo (MOD), AND, OR, and XOR. Data processing order : (at the reading) 4
Data type conversion function	This converts the data type of item to the specified data type. This function supports Array type as well. Data processing order : (at the reading) 5,(at the writing) 1.

Dead band setting	Specify the range that is not considered as a value change. CaoSQL regards that the value is not changed if the value fluctuates within this range. If there is an operation (for example, event notification and item link) that is link to a value that the dead band setting is used, when you change the dead band setting value, the behavior of the operation will change accordingly. This function is effective to integer type data that is not an array.
Valid range setting	Specify the range that is regarded as a value change. CaoSQL regards that the value is not changed if the value fluctuates out of this range. If there is an operation (for example, event notification and item link) that is link to a value that the valid range setting is used, when you change the valid range setting value, the behavior of the operation will change accordingly. This function is effective to integer type data that is not an array.
Chattering disregarding function	To disregard the chattering, this function updates the value change only when the value is sampled in predetermined time.
Item link function	When an item's value is updated, if the item links to the different item, the updated value is sent to the link destination item. This function works as if these two items are physically wired with I/O in software.
Inactivation function	For each item, specify whether to write/read the item in the sampling interval of the controller.
Array type item function	When data reading, each array element is transmitted to each item being linked. When data writing, the array elements are overwritten by the linked item's values . This function is practical for bit data (binary digit) operation.
Alias function	With this function, both normal and array type items can be registered with different names. Values are linked to the newly created items. You can process data as well as send item's value registered as alias to others.

3.2. Event notification function

Event notification function realizes the event-driven programming, therefore polling processing on the client side is not required. As a result, you can create efficient program. In CaoSQL, since an event is issued in the sampling thread, sampling may delay depending on the event handler's processing on the client side. Also, if the event handler calls CaoSQL's function recursively, processing may be locked at times. To prevent that, set the event handler in the client program asynchronous as much as possible.

This sampling thread is generated by each controller. The event issuance can be cancelled in each controller by using CaoSQLConfig tool setup. Also, you can cancel the event issuance in each item, so, cancel the event issuance when event is not used.

The value change of the following 17 data types can be detected;

VT_EMPTY, VT_NULL, VT_BOOL, VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, VT_UI4, VT_R4, VT_R8, VT_CY, VT_DATE, VT_BSTR, VT_ERROR, VT_DISPATCH, and VT_UNKNOWN.

For Array data, when [Compare Arrays] is enabled, each element of the array are compared. If it is disabled, values are deemed to change at any time. However, this function may degrade performance since each element of array are compared.

3.2.1. OnChangeItem event

OnChangeItem event is issued when the value of the registered item changes. There are two arguments; item name (BSTR) and value (VARIANT). The item name is not case-sensitive. Item names in each controller are not overlapped.

3.2.2. OnChangeState event

OnChangeState event is issued when the state value of the registered controller changes. This event has one argument; Status value (long).

For about state value, refer to Table 3-2 The controller's status value list and Appendix A.2.

Succeeding to the status change, the client application can execute processing in every scanning.

Table 3-2 The controller's status value list

Status value	State explanation
0	Thread not initialized
1	Thread activated
2	Thread deactivated
3	Thread error issued
4	Complete the thread

3.3. Trigger function

The trigger function assigns an action to a sampling thread when specified condition is satisfied. For example, some items in a controller need the sampling only when a certain condition is satisfied. In this case, setting trigger will reduce overhead, for example, if a trigger is not set, unnecessary items are sampled every time. In addition, this will prevent other negative effects on sampling interval.

The trigger function has the following items. (for setting method, refer to "[CaoSQLTools user's guide](#)".)

Table 3-3 Trigger function setting item

Item	Content	
Trigger item	CaoSQL item name that monitors condition	
Action	Start	Start sampling of all items
	Stop	Stop sampling other than the trigger items
	1 Time	Sample all items only once.
Conditional expression	==, !=, <, <=, >, >=	
Threshold	Value to be compared with the trigger item	

Note that operation of *Stop* of the *trigger function* is quite different from *Stop* of *CaoSQLController method*. In *Stop* action of the *trigger function*, the thread itself is not stopped, and sampling of the specified trigger item keeps running.

On the other hand, *CaoSQLController::Stop* stops the thread itself. This is effective when the process is limited by time (for example, start/stop the system according to the operation start time).

3.4. DDE server function

With this DDE server function, users can obtain CaoSQL item's value from DDE client applications, such as Microsoft Excel, without programming.

CaoSQL supports the DDEML-used DDL server function only for the value reading. Since CaoSQL supports hotlink, CaoSQL can notify the DDE client of the value change. To use DDE function of CaoSQL, specify a necessary item from the following information.

Table 3-4 DDE server information

Item	Setting	Remarks
DDE server name	CaoSQL	Enter "CaoSQL" at any time.
Topic name	Controller name	CaoController name that has been configured by CaoSQL
Item name	Item name	Item name registered in CaoController name that has been configured by CaoSQL. Limited to <u>Read</u> attribute.

The following sample shows when Excel2000 is configured as DDE client.

(1) Enable DDE server function

Start CaoSQLConfig. From the menu bar, click *Action*, click *Option*, select *API* tab, select *DDE Server*

checkbox, and then click OK button.

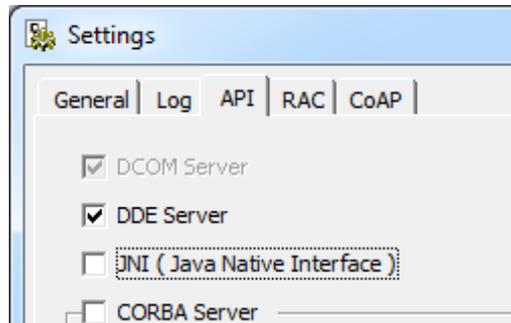


Figure 3-1 Setting of DDE server function

(2) Start CaoSQL.

Start CaoSQL by using CaoSQLLauncher. However, if you start CaoSQL as a service, from the CaoSQL_Properties (Local Computer) window, on the Log On tab, select “Allow service to interact with desktop” checkbox. If your computer is Windows XP, click [Administrative Tools] – [Services] – [Properties] – [Logon] tab, if your computer is WindowsNT4.0, click [Service] – [Startup] – [Logon] tab, if your computer is Windows2000, click [Service] – [Properties]– [Logon] tab.

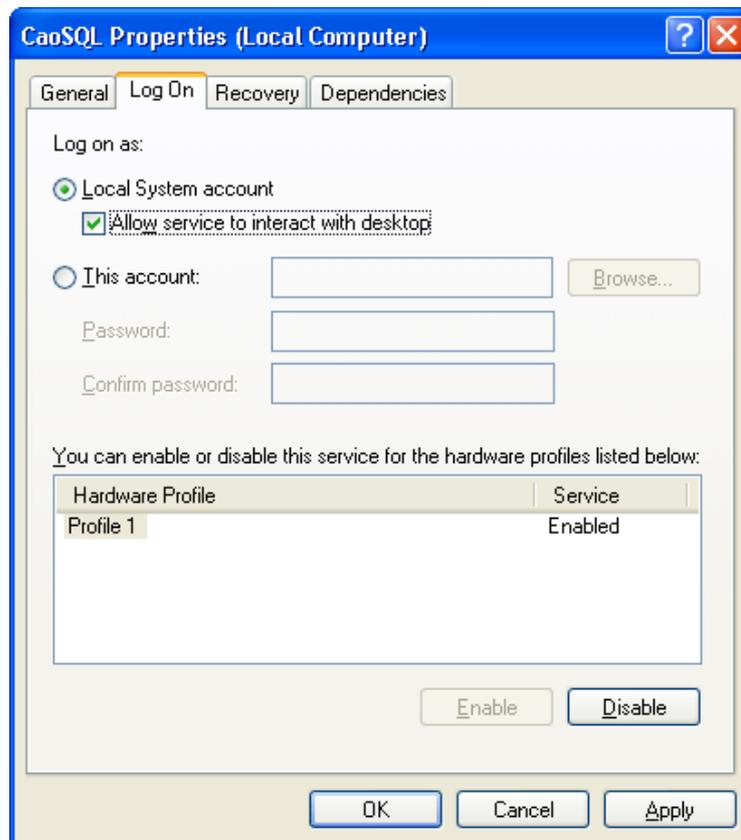


Figure 3-2 Service setting with Windows XP

(3) Start Excel, and register an item.

CaoSQLConfig equips a function that copies the registered DDE string and save it to the clipboard so that users can easily register items when using Excel. (Refer to the copy of the DDE character string of "Figure 3-3".) The function to copy the DDE registration character string to the clip board to register the item in CaoSQLConfig easily when using it by Excel is provided. (Refer to Figure 3-3 Copy the DDE character string.) Paste the copied DDE character string to Excel sheet in order to refer items on the DDE server.

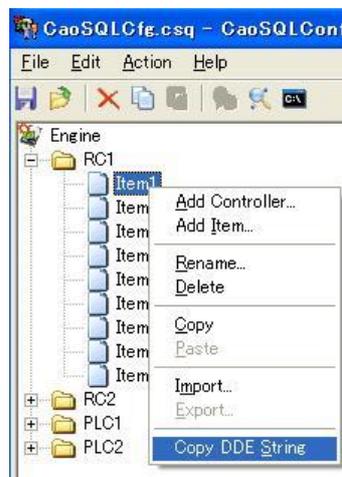


Figure 3-3 Copy the DDE character string

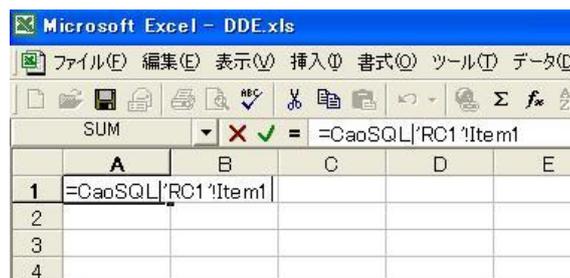


Figure 3-4 Paste the DDE character string to Excel.

This completes necessary settings. If Excel2000 is a DDE client, DDE server is hot-linked. So if an item's value changes, the change is informed to DDE client and the value is reflected in the linked cell.

3.5. Network DDE server function

This function allows to use DDE server function on a network. CaoSQL offers network DDE function by using DDEShare program (ddeshare.exe).

For example, user can obtain CaoSQL data that is running on the server computer with network DDE by using client computer's Excel. The following shows setup procedure when network DDB is used.

(1) Set DDE Share on

To use network DDE, you need to configure "DDE Share on". From the start menu, click Run, and then enter DDEShare.exe to start.

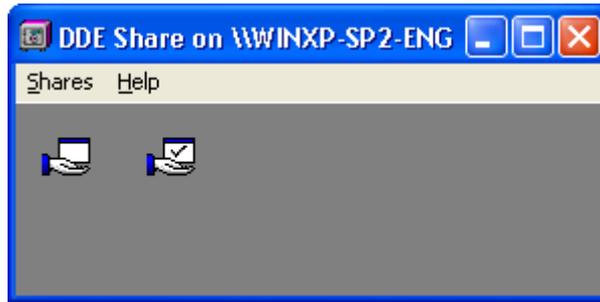


Figure 3-5 The main window of DDE Share on

Configure "DDE Share on" for both the server and client. To enable this configuration, you need to log on with "Administrator" before configuration.

(2) Create DDEShare

From the menu of "DDE Share on", select "DDE share". DDE share list is displayed



Figure 3-6 DDE common dialog

To set a new sharing, select “Add a Share...”. If you want to edit an existing sharing, click DDE Share Properties.

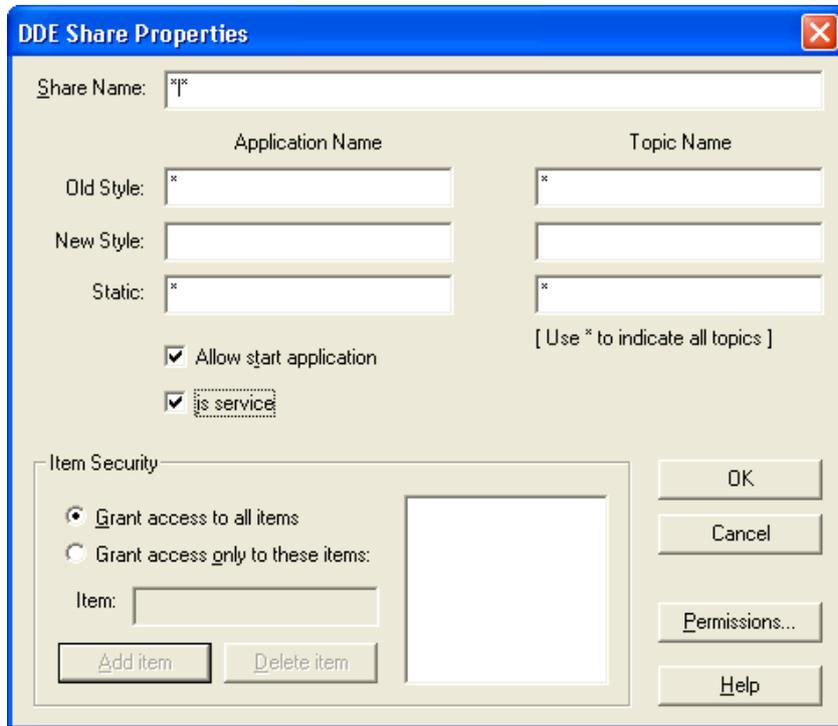


Figure 3-7 Additional dialog of DDE sharing

To set an access control, click [Permissions] button. To use Network DDE, you need to configure “Trust Share...” from the client computer. Note that you need to log on with *Administrator* when you configure this setting.

(3) Use DDE on the client application.

On the client application (in this case, Excel), obtain data from the server by using DDE. To use DDE, enter DDE character string into the cell of Excel.

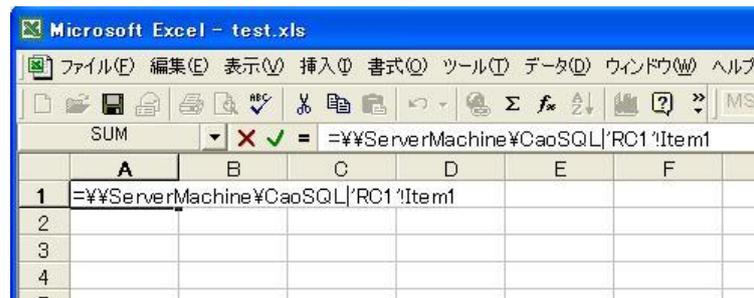


Figure 3-8 Entering DDE character string

The following shows the character string form when network DDE is used.

“= ‘\\Machine name\Application name’ | ’Topic name’! ’Item name’”

Note that this form changes depending on the DDE sharing setting (see below).

[Reference]

The above-mentioned setting example is applied to all applications. If you want to enable only one item (in this case, item name “DS1”) under CaoSQL’s controller, configure as follows.

- Shared name:「NetDS1\$」
- Application name (static) "CaoSQL", (Old style/New style): Blank
- Topic name (static) "DS1", (Old style/New style): Blank

Set DDE character string as follows. NDDE\$ is a fixed value.

“\\ < Machine name > \NDDE\$!NetDS1\$!Item1”

[Notes]

To run the network DDE, there are some tips to confirm other than above. Please refer to the following site.

<http://www.angelfire.com/biz/rhaminisys/ddeinfo.html>

Also, at least, the following points must be confirmed.

- (1) CaoSQL runs on the server computer.
- (2) Network DDE and Network DDE DSDM service are run on both the server and client computers.
- (3) Network DDE works at least in the local computer.

3.6. Log function (Error logging)

This function allows to log data while the specified operation runs, such as creating/deleting objects, start/stop of caoSQL. User can select an output destination from the followings; console, message box, event viewer, debug viewer, and text file. The output destination is configured by CaoSQLConfig.

Log-output timing of CaoSQL is as follows.

- CaoSQL initialization error or termination error

- Controller's sampling thread, start/end, run/stop
- Parameter error of controller or item is found
- CaoController object creation failure
- CaoController startup time-out
- CaoVariable object creation, value obtainment/setting failure
- Database connection failure in History function
- Database writing failure in History function
- DDE server startup failure
- Advise fails in DDE hot link function
- Trigger is executed

In addition, if logging target issues an error, error log output may occur repeatedly. To prevent that, from the menu bar of CaoSQLConfig, click Action, click Option, select Log tab, and then remove the check from the checkbox of "NOT repeat the same error". Note that the error-state is reset once processing completes normally at least one time. If the same error occurs again after the error-reset, this error will be output to the log. This mechanism is practical when ;

- value acquisition/setting of CaoVariable fails, or,
- database writing is failed because of the History function.

3.7. Cascade connection function of CaoSQL

The following explains how to construct the CaoSQL distributed computing environment.

CaoSQL distributed computing environment enables to collect necessary data in one computer efficiently.

Similar function can be achieved by using interface of CAO as well though, using CaoSQL, which works as a middleware, brings the following benefits.

- Since CaoSQL provider caches controller's data, client can refer the cached data, leading to reduce the load on the robot controller.
- By utilizing CaoSQL interface, users do not need to create dedicated programs for respective client application.

The following explains how to establish the CaoSQL distributed computing environment with the following four providers.

- CaoSQL provider
- RAC provider
- e-CAP provider
- CoAP provider

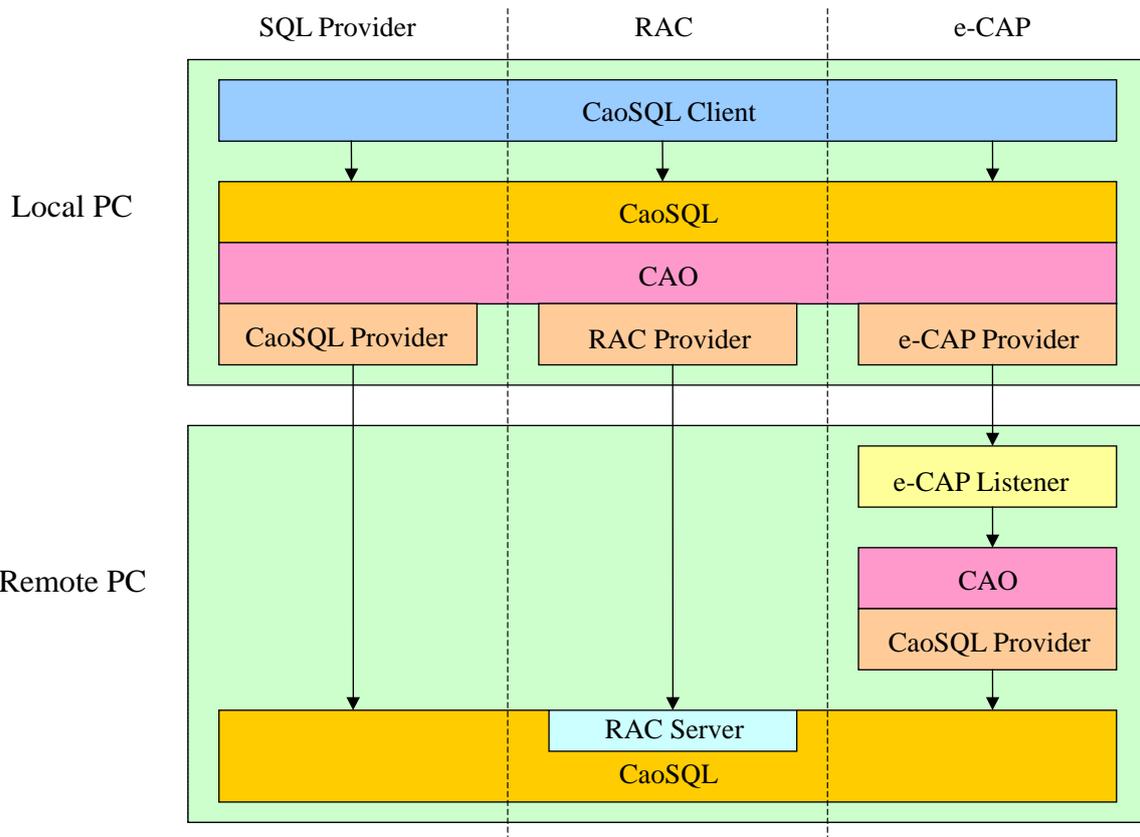


Figure 3-9 CaoSQL distributed computing environment

3.7.1. CaoSQL provider

CaoSQL provider can obtain one CaoSQLController running on a certain computer by using CAO interface, and also obtain CaoSQLItem as a CaoVariable. For information on how to use CaoSQL provider, refer to "[CaoSQLTools user's guide](#)".

Since CaoSQL provider accesses CaoSQL on a remote computer by using DCOM, to realize cascade connection, you need to configure security settings by using dcomcnfg.exe tool that is accompanied by DCOM module of Microsoft⁶. This setting aims to realize mutual communication of CaoSQLs on different computers and to prevent multi-booting of CaoSQL.

⁶ For information on how to use dcomcnfg.exe., refer to "[ORiN2 programming guide](#)".

To establish the connection, configure the authentication level with dcomcnfg.exe. Since CaoSQL provider does not use CaoSQL event, set the succeeding layer to "None"(see Figure 3-10)). However, if you use Windows 9x series, you must start up CaoSQL beforehand because CaoSQL on this computer cannot be started from remote computer due to its DCOM specification.⁷

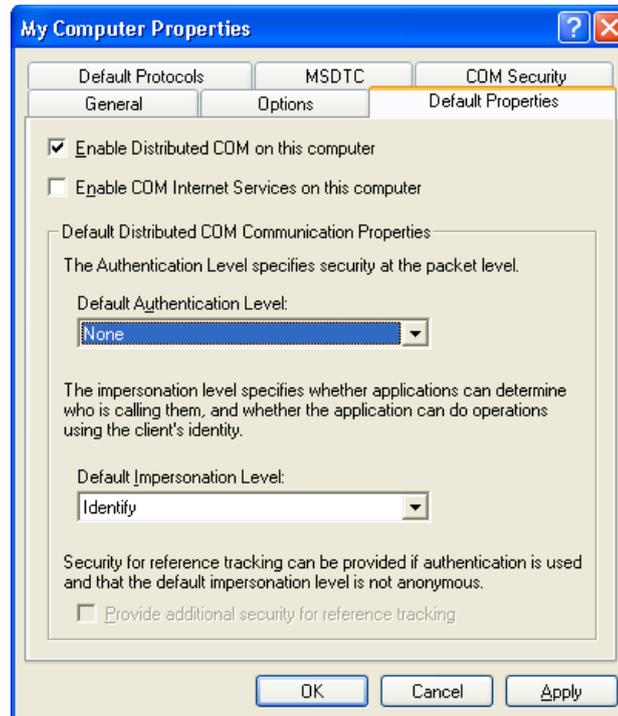


Figure 3-10 dcomcnfg - "Default Properties"

⁷ In this case, CaoProvExec tool is useful. Please refer to "[CAO provider making guide](#)" for CaoProvExec tool.

Next, to avoid multi-booting of CaoSQL process, clarify the user of the processing. From "CaoSQL Engine Class Properties", click Identity tab, and then select "The interactive user".⁸ In this setting, the multi-booting of CaoSQL never occurs because CaoSQL process is always started by the process of the user who logs on. Similarly, you can explicitly integrate the user of CaoSQL process by selecting "This user".



Figure 3-11 dcomcnfg - "CaoSQL Engine Class Properties"

3.7.2. RAC provider

RAC provider is used to communicate with a remote computer by sending and receiving RAC commands. For information on use of RAC provider, refer to "[RAC provider guide](#)".

Using RAC server on the remote computer's CaoSQL allows to establish cascade connection.

3.7.3. e-CAP provider

e-CAP provider is used to communicate with a remote computer by using HTTP. The cascade connection is achieved by this provider across Firewalls. TTP.

For information on how to use e-CAP provider, refer to "[e-CAP provider guide](#)".

3.7.4. CoAP provider

CoAP provider is used to communicate with a remote computer by sending and receiving packets with

⁸ If [The launching user] is selected, when a log-on user has already started CaoSQL, the communication will fail because two different users (the log-on user and the user who has started CaoSQL from the upper layer (in this case, User0)) will execute two different CaoSQL processes. This also doubles the load on the computer.

CoAP protocol. For information on how to use CoAP provider, refer to "[CoAP provider guide](#)".

3.8. History function (Data logging)

History function is a data logging function that records Item's value in the database when the value changes or when Snapshot method is explicitly called⁹.

To specify the database for History, use CaoSQLConfig tool (See 4.2.1.1 for details). If you set the maximum record number of History, History will stop recording further data when the number of record exceeds the maximum¹⁰.

You can choose database; however, you need to create a database and table according to your needs since CaoSQL does not create a database and table newly. [Table 3-5](#) shows the structure of database. A new table is created by executing the following SQL sentence.

```
CREATE TABLE caosql_history
(controller_name varchar(32) NOT NULL,
item_name varchar(32) NOT NULL,
item_value varchar(256),
item_vartype short NOT NULL,
item_time datetime NOT NULL,
item_time_ms long NOT NULL,
flags short NOT NULL)
```

This data log function can be turned off in each Controller or Item. In CaoSQL, independent thread performs sampling by each controller; however, all log outputs are stored in one common database and are serialized at that timing. Therefore, the performance of the sampling may be degraded. Since the log output operation requires a large overhead, it is recommended not to output unnecessary controllers or items.

Table 3-5 Structure of history table

Field name	Data type	Length (Byte)	Remarks
controller_name	varchar NOT NULL	32	Controller name that a target item belongs. If the data size is 32 bytes or more, an error occurs.
item_name	varchar NOT NULL	32	Target item name If the data size is 32 bytes or more, an error occurs.
item_value	varchar	256	Convert the value after change into a character string, and then record it. (The data of 256 bytes or more is recorded up to 256 bytes.)

⁹ When you use Snapshot method, turn off the [Auto History] function in order to distinguish the automatic recording and other recording. By turning off the [Auto History], data will be recorded in the database only when Snapshot method is called.

¹⁰ The overhead increases because the inspection is executed in every log-output timing. For normal use, turn OFF the record number restriction, and set the capacity limit on the database engine side.

	(Note) If the data type of all data to record is the same, you can set the data type as the data. In this case, enable the option of "No Conversion To BSTR" so that the character string is not converted by CaoSQL.		
item_vartype	short NOT NULL	2	Record the type of value after change by numeric value ¹¹
item_time	datetime NOT NULL	8	Record time that the value changes.
item_time_ms	long NOT NULL	4	Record time (ms) that the value changes.
flags	short NOT NULL	2	Record the type of the record. 0:History data and 1: The latest data ¹²

CaoSQL executes the processing of the database by using ADO (ActiveX Data Object). Therefore, if the database system equips ADO provider (OLE-DB provider), the database could be handled by CaoSQL. The following databases have been confirmed to work with this function.

SQLServer, Oracle, Access, Excel, CSV file, MySQL, PostgreSQL

Table 3-6 Content of database set list shows the connection strings for the general database systems.

Table 3-6 Content of database set list

Database	Table name	Specified option	Content
SQL Server	Table name	Provider	"SQLOLEDB.1"
		Data Source	SQL server name
		Initial Catalog	Database name
		User ID	User ID of database
		Password	User's password
Oracle	Table name	Provider	"MSDATAORA"
		Data Source	Oracle server name
		Initial Catalog	Database name
		User ID	User ID of database
		Password	User's password
Access97	Table name	Provider	"Microsoft.Jet.OLEDB.3.51"

¹¹The numerical value of each type is as follows: VT_EMPTY=0, VT_I2=2, VT_I4=3, VT_R4=4, VT_R8=5, VT_CY=6, VT_DATE=7, VT_BSTR=8, VT_DISPATCH=9, VT_ERROR=10, VT_BOOL=11, VT_UNKNOWN=13, VT_I1=16, VT_UI1=17, VT_UI2=18, and VT_UI4=19.

¹² In the "Table Type" setting on CaoSQLConfig, if "History" is selected, only the History data is recorded. If "Current Value" is selected, only the latest data is recorded. If "History & Current Value" is selected, both the History data and the latest data are recorded.

		Data Source	File name (*.mdb)
Access2000	Table name	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	File name (*.mdb)
Excel2000 ¹³	[<sheet name>\$] or [<sheet name>\$<start cell>:<end cell>] ¹⁴ Example:[Sheet1\$A1:G31]	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	File name (*.xls)
		Extended Properties	"Excel 8.0" ¹⁵
CSV file ¹⁶	CSV file name ¹⁷	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	Folder path of the CVS file
		Extended Properties	"Text"
MySQL	Table name	Provider	MSDASQL
		Misc.	"Driver={PostgreSQL Unicode}; Server= Server name; Port=Port No.; Database= Database name; UID= User ID of database; PWD= User's password"
PostgreSQL	Table name	Provider	MSDASQL
		Misc.	"Driver={MySQL ODBC 8.0 Unicode Driver}; Server= Server name; Port=Port No.; Database= Schema name; UID= User ID of database; PWD= User's password"

3.8.1. System information recording

In History function, <controller_name> records the controller name that performs sampling, and <item_time> and <item_time_ms> records the duration of sampling. <flag > is the same as the record of item value. Type to be recorded in the record and field values other than above, refer to Table 3-7.

Table 3-7 System information list

item_name	item_value	item_vartype	Description
\$\$START\$\$	-	0 (VT_EMPTY)	Sampling start

¹³ For a sheet, enter a title row that is entered a field name listed in Table 3-5

¹⁴ You need to enclose the specified sheet name and the cell range by square brackets "[]".

¹⁵ Enter a blank between "Excel" and "8.0".

¹⁶ For a CSV file, "UPDATE" and "DELETE" cannot be executed. In the first row of the CSV file, enter a column name with comma delimiters

¹⁷ Specify a CSV file name. Do not specify a file path of a CSV file.

\$\$STOP\$\$	-	0 (VT_EMPTY)	Sampling stop
\$\$BEGIN\$\$	Update count from the startup.	3 (VT_I4)	Start the value update
\$\$END\$\$	Update count from the startup	3 (VT_I4)	Stop the value update
\$\$OPEN\$\$	-	0 (VT_EMPTY)	Open a database
\$\$CLOSE\$\$	-	0 (VT_EMPTY)	Close a database
\$\$SWITCH\$\$	-	0 (VT_EMPTY)	Switch database from primary to secondary

3.8.2. Converting character string of an item value

An item value is recorded into the record after converted to a character string.

An array data is converted into a character string with delimited by commas.

When you limit the data type of <item_value> field (for example, because the recorded data type is only one), disable this function.

3.8.3. Saving binary data into a file

In History function, binary data can be saved in a file.

Once a file is saved, the saved file name is recorded in the item value recorded in the record.

The saved file name is determined based on the fields of the record as follows.

History data <controller_name>_<item_name>_<item_time>_<item_time_ms>.dat

The latest data <controller_name>_<item_name>_Current.dat

Note that delimiters are not added to the value of <item_time>.

(Ex.: 2015/07/01 10:11:12 → 20150701101112)

A destination folder of file saving is created in the same location of the CaoSQL setting files. The saved folder name will be the character string where the extension is removed from the setting file.

Since this function has no restrictions on the maximum file count or file size, to limit the maximum file count or disk capacity to use, consider the followings.

- The maximum record count of the History function
- Size limit of the database
- Drive usage of the disk quota

3.9. Secondary DB switch function

One database can set two databases; a primary database and a secondary database. If any database server error disturbs recording while item's values are being recorded in the primary database by using a History

function, the database automatically switches to the secondary database in order to keep recording. This will prevent any missed data for recording.

A typical usage of this function is to set the database server on the network to the primary database (for example, SQL-Server), and to set the local database server to the secondary server (for example, Access). Doing so, all data will be recorded without fail, even if any network error occurs.

The switch of database is allowed to one direction only (switching from the primary database to the secondary database). Once the database is switched to the secondary database, it is impossible to be back to the primary database. Since the secondary database is intended for the emergency use, you need to maintain the database server at any time to prevent that change.

Also, even if an error occurs when an arbitral SQL statement is executed from the client program by using Execute method (a function that executes an arbitral SQL statement) on CaoSQLHistory class, the database is not changed automatically. This is because the system cannot determine whether the cause of error is the database server or the SQL statement

3.10. Array element extraction function

If the value of the item is an array, you can extract necessary elements only. Delimit the index numbers of an element you want to extract with commas, enclose them with brackets, and then add it to an item name. Up to three dimensions can be specified.

Example) Item1(2,3)

The following 13 data types can be extracted. VT_BOOL, VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, VT_UI4, VT_R4, VT_R8, VT_CY, VT_DATE, VT_BSTR, and VT_VARIANT.

(Note) This function enables to perform mask processing for each array element, to perform data processing such as BCD conversion, and to allocate different link destination by each element. However, if these processing is not required and you simply want to obtain an element from an array, it may be efficient to obtain an array as-is, and then extract a necessary element on the client application side.

3.11. Data masking function

This function masks values of an item. By setting mask values on bits you want to extract, masking will be executed automatically. Masking will be done for both the reading and writing. If this is executed with BCD conversion function and Binary operation function simultaneously, it will be processed in order of Masking processing >> BCD conversion >> Binary operation.

This function is available to signed or unsigned Integer type of 1 byte, 2 bytes, and 4 bytes (not array type). Concretely, the following six data types are available: VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, and VT_UI4.

3.12. BCD conversion function

Convert item's value into BCD (Binary Code Decimal). Conversion is done for both the reading and writing. If this function is executed with Masking function and Binary operation function simultaneously, it will be

processed in order of Masking processing >> BCD conversion >> Binary operation.

This function is available to signed or unsigned Integer type of 1 byte, 2 bytes, and 4 bytes (not array type). Concretely, the following six data types are available: VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, and VT_UI4.

3.13. Binary operation function

Using a binary operation function, you can perform a binary operation on a value that the own item obtains and a value of another item, and the calculation result will be cached as an own item's value. It means the following processing will be executed¹⁸.

<Value of own item> <Operator> <(another)Item>

This conversion is executed only at the time of reading. For <Operator >, the following eight types are available: Additive (+), Subtraction (-), Multiplication (*), Division (/), Surplus calculation (MOD), logical product (AND), logical add (OR), and exclusive-OR (XOR). If this function is executed with Masking function and BCD conversion function simultaneously, it will be processed in order of Masking processing >> BCD conversion >> Binary operation.

The format of <(another) Item > is

"[<Controller Name>\]<Item Name>" or "#<Constant>"

When a controller name is omitted, it is deemed that the controller of own item is specified. A constant is treated as a double precision real number. For example, you can enter values in the form of "#100"(decimal)" or "#&H10000"(hexadecimal)".

For example, assuming that the condition "I/O1 of the part number X turns ON" is specified. If you set an AND operation with IO1 for the item number X, the upper client can judge the condition by using one item only.

The implementation of this binary operation utilizes the function of Table 3-8 Internal function of binary operation function, which is of the oleaut32.lib. For details information, such as operation result between different types, refer to documents of Microsoft.

For one item, binary operation is available only. However, complicated operations will be available if you combine binary operations with the link function so that all complicated operations are decomposed into binary operations. For details, refer to 3.17.

Table 3-8 Internal function of binary operation function

	Operation	Notation	Return value	Internal processing function
1	Addition	+	(operation result)	VarAdd()
2	Subtraction	-	(operation result)	VarSub()

¹⁸There is a possibility that the arithmetic processing is infinitely done when it makes a mistake in the setting, and set it carefully, please.

3	Multiplication	*	(operation result)	VarMul()
4	Division	/	(operation result)	VarDiv()
5	Surplus of division	MOD	(operation result)	VarMod()
6	Logical product	AND	(operation result)	VarAnd()
7	Logical add	OR	(operation result)	VarOr()
8	Exclusive-OR	XOR	(operation result)	VarXor()
9	Left shift	<<	(operation result)	<<
10	Right shift	>>	(operation result)	>>
11	Equal.	=	True / False	VarCmp()
12	Not equal.	<>	True / False	VarCmp()
13	Less than or equal to	<=	True / False	VarCmp()
14	Less than	<	True / False	VarCmp()
15	Greater than or equal to	>=	True / False	VarCmp()
16	Greater than	>	True / False	VarCmp()
17	Concatenation	CAT	(operation result)	VarCat()
18	Integer division	IDIV	(operation result)	VarIdiv()
19	Implication	IMP	(operation result)	VarImp()
20	Power	POW	(operation result)	VarPow()
21	Absolute	ABS	(operation result)	VarAbs()
22	Integer portion	FIX	(operation result)	VarFix()
23	Integer portion	INT	(operation result)	VarInt()
24	Variant negation (Sign inversion)	NEG	(operation result)	VarNeg()
25	Bit inversion	NOT	(operation result)	VarNot()
26	Convert radian to degree	RAD2DEG	(operation result)	-
27	Convert degree to radian	DEG2RAD	(operation result)	-
28	String length	LEN	(operation result)	-
29	Substring (Left)	LEFT\$	(operation result)	-
30	Substring (Right)	RIGHT\$	(operation result)	-

3.13.1. The second operation function

You can execute the second operation¹⁹. The format and the operation processing are the same as above. The second operation is done for the result of the first operation. Using the second operation function allows to execute the following conversions.

¹⁹ This function is available from Ver.1.8.0. or higher.

$X \leftarrow AX + B$ ' the first operand='*', the second operand ='+'

$X \leftarrow (AX)^B$ ' the first operand='*', the second operand ='POW'

3.14. Data type conversion function

This function converts an item's data type into specified data type. It supports multidimensional array. You can specify data type to convert for reading and writing, respectively.

Available data types for conversion are the following 13 types. VT_BOOL, VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, VT_UI4, VT_R4, VT_R8, VT_CY, VT_DATE, VT_BSTR, and VT_VARIANT.

3.15. Dead band setting function

Specify the range that is not considered to be a change of the value by entering the minimum and the maximum values. Since the value change occurred in this range is not considered to be a change of the value, a function that accompanies the value change is not executed. Functions that accompany the value change are the following four functions.

- Event notification function (OnChangeItem event)
- Item link function
- Trigger function
- History (Data logging) function

This function is available to signed or unsigned Integer type of 1 byte, 2 bytes, and 4 bytes (not array type). Concretely, the following eight data types are available : VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, VT_UI4, VT_R4, and VT_R8. When the data type changes, it is judged that the value changed unconditionally.

3.16. Valid range setting function

Specify the range considered to be a change of the value by entering the minimum and maximum values. Since the value change occurred outside this range is not considered to be a change of the value, a function that accompanies the value change is not executed. Functions that accompany the value change are the following four functions.

- Event notification function(OnChangeItem event)
- Item link function
- Trigger function
- History (Dataroging) function

This function is available to signed or unsigned Integer type of 1 byte, 2 bytes, and 4 bytes (not array type). Concretely, the following eight data types are available : VT_I1, VT_UI1, VT_I2, VT_UI2, VT_I4, VT_UI4, VT_R4, and VT_R8. When the data type changes, it is judged that the value changed unconditionally.

3.17. Item link function

The item link function is a function to transmit an item's value to a link destination item when an item's value changes. To use this function, an item needs to link with the link destination item beforehand. This function performs as if items are linked by soft-wired connection.

For example, if item I1 of RC1 is linked with item I1 of RC2, when I1 of RC1 changes to 1 from 0, CaoSQL detects the change and writes 1 into item I1 of RC2.

Also, you can obtain the link destination information that is set by CaoSQLConfig from CaoSQLItem interface. The format is as follows:

[<Controller Name>]<Item Name>

When a controller name is omitted, it is deemed that the controller of own item is specified. To specify multiple items, delimit them with commas.

In the default setting, the same value will be transmitted to all link destination items. If the array element distribution link function (Extract Linking option) is enabled and the item is array data, extract respective array elements and then transmit these values into link destination's items in order.

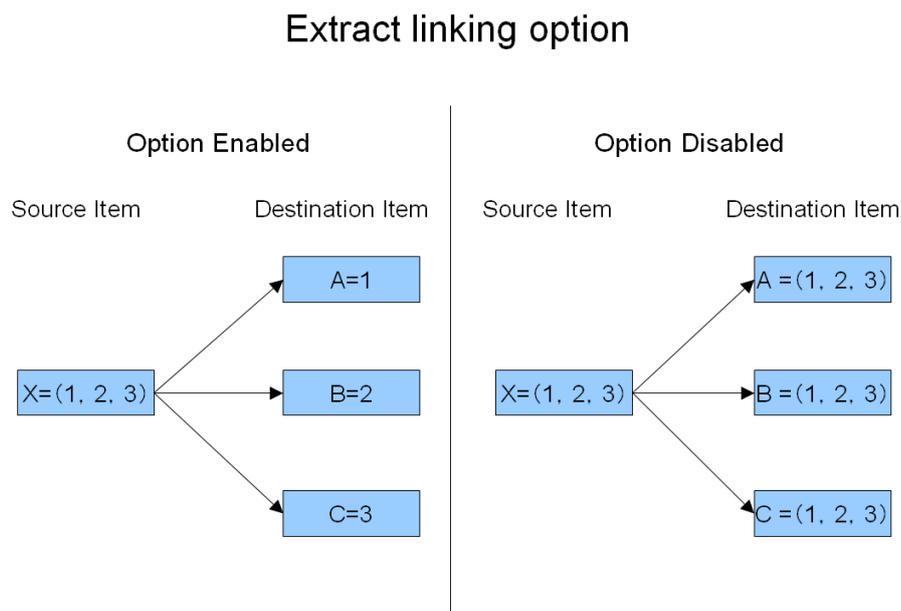


Figure 3-12 Array element distribution link function

This link function is handy in terms of value transmission without special programming though, once you set a wrong link, infinite data transmission might occur. For example, under the conditions of Figure 3-13 Infinite loop with mutual link, if the resource α and the resource β take the different values at the same time, processing will loop infinitely because these two are linked mutually. This applies to the case when you program with CaoSQL interface²⁰.

²⁰ Same as a physical IO connection, infinite loop for a CaoSQL soft-wired connection is abnormal.

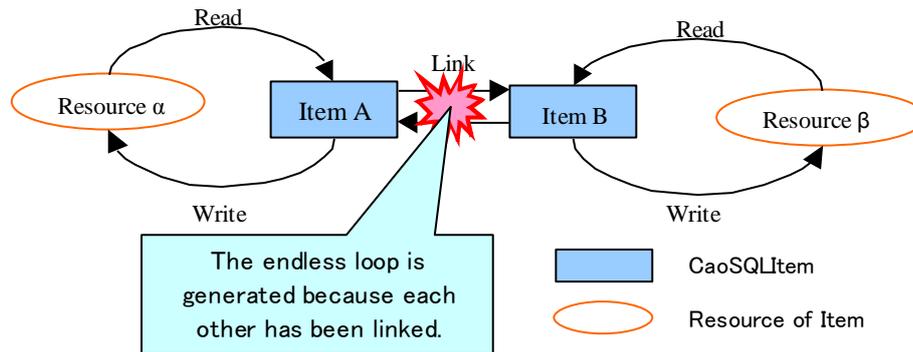


Figure 3-13 Infinite loop with mutual link

3.17.1. Selecting Link destination property

In the default setting, the Value property of the link destination item is selected. You can change this property from [Details] of Item tab. Available properties are as follows: Value (default), ID, Activation, and LinkActivation.

For example, assuming that a variable object A (CaoVariable) is a provider that obtains an array element specified by ID property. By using this link function, a different variable object B can specify the ID property of variable object A, so you can change the element number of variable object A by changing the variable object B

3.18. Inactivation function

If this function is enabled, reading and writing of items are not performed at the sampling period of a controller. To enable this function when CaoSQL is executed, enable the Activation property. In the default setting, this function is invalid.

3.19. Array type item function

If Array type item function is used, when an array type item is read from an external device, such as PLC, the item's contents are decomposed and transmitted to the linked items. To write an array type item's values into an external device, the values of the linked items are transmitted to the array type item, and the array is transmitted to the device.

However, the order of items that the link are set might not be as intended. When you use this function, set the link in ascending order by using the sort function.

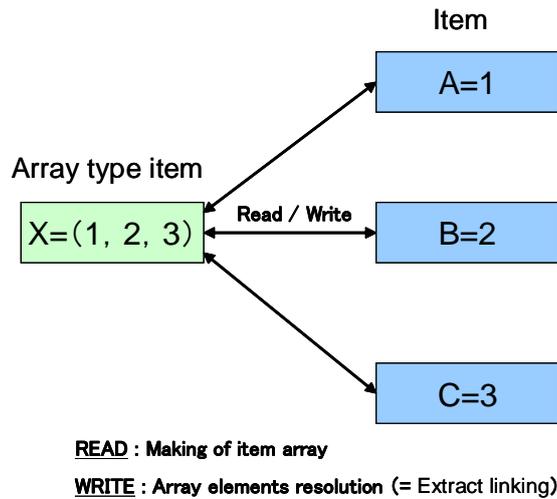


Figure 3-14 Array type item function

3.20. Alias function

As 2.2 Object model explains, you can set each alias function for respective controllers and items. Alias function is a function to register an already registered item with a different name, and to transmit only its values. Transmission information to be set to the alias can be obtained by CaoSQLItem interface.

The format is as follows:

[<Controller Name>\
<Item Name>

To transmit an array type item's value to which the link cannot be set, use this alias function.

3.21. RAC Server function²¹

RAC server function enables to exchange data with CaoSQL via the network by using RAC provider. Two or more RAC servers can be set.

To set RAC server function, from the menu of CaoSQLConfig.exe, click [Action] -> [Option], and then select [RAC] tab (Refer to 4.2.2.3). Specify the option character string of AddController of RAC provider for setting. Enter the setting of server mode since only the server function has been implemented at present.²² For more details about RAC provider, refer to [RAC Provider Guide](#).

The following image shows the example of the setting window of RAC server function.

²¹ The development of this function was supported by Japan Society for the Promotion of Machine Industry, JSPMI.

²² Concretely, set the connection parameter to **Ethernet**, the Ether parameter to **Server**, and **Unicode conversion** of the packet parameter to **ON**. Because the server mode is automatically set to the asynchronous mode, Sync option is not related.

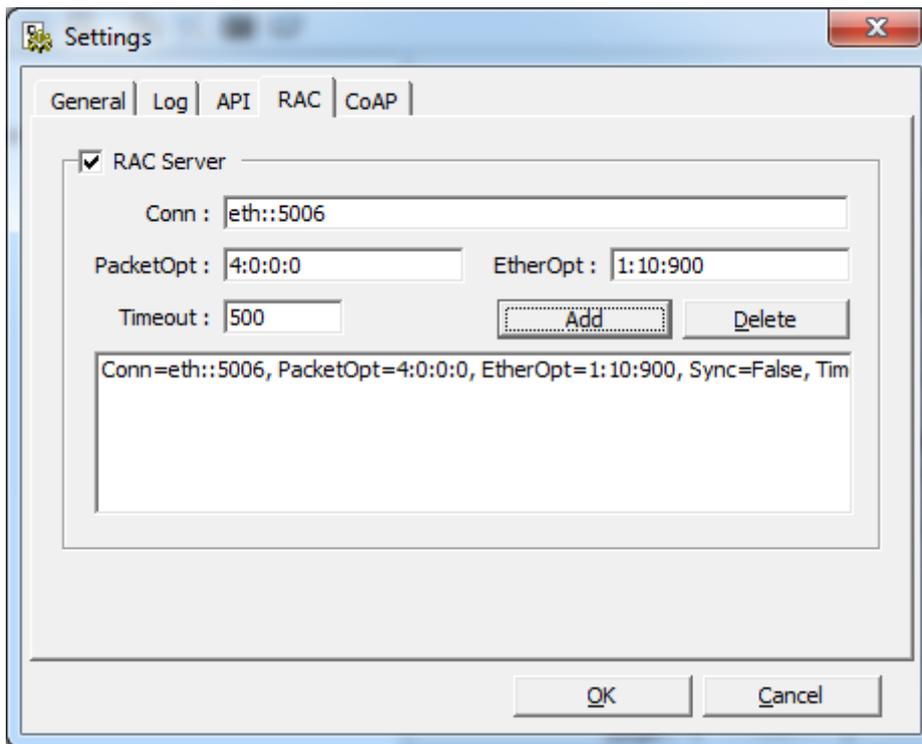


Figure 3-15 Example of setting RAC server

The following introduces available RAC commands.

3.21.1. GET command

The format of GET command is as follows. ²³

GET :[<controller name>][<item ID>]:[<item name>][!<property name>]:

Acquire the property value of a specified item. When <item ID> is entered, ID is set (put_ID) immediately before the acquisition. ²⁴

When the controller name is omitted, the default controller name is used.

For about the property list of available engines and controllers, refer to [Table 3-9](#).

Table 3-9 RAC server function Engine/Controller property list

Item name	Description	GET	PUT
ControllerNames	Obtain the controller name list. This item is available only when "\$ENGINE" is specified.	✓	-
ItemNames	Obtain the item name list of the specified controller.	✓	-

²³ (note) You need to enter two colons (::) between the controller name and the item name, and one colon (:) after the item name.

²⁴ In the automatic sampling active, please note that there is a possibility that the value has not been updated, too, even if ID is set.

Description	Obtain the value of Description property of the specified controller.	✓	-
State	Obtain the value of State property of the specified controller.	✓	-

[Table 3-10](#) shows the property list of available items. If the item name is omitted, Value is used

Table 3-10 RAC server function Item property list

Property name	Description	GET	PUT
Value	Item value	✓	✓
Attribute	Attribute value	✓	-
Datetime	Timestamp (date)	✓	-
Description	Description	✓	-
Microsecond	Timestamp (microsecond)	✓	-
State	State	✓	-

Example 1) GET:Ctrl1::Item1: // Obtain the value of Item(Item 1) of Controller(Ctrl1).

Example 2) GET::Item2!Value: // Obtain the value of Item (Item2) of the default controller.

Example 3) GET:Ctrl2::Item1!Attribute: // Obtain the attribute value of Item(Item1) of Controller (Ctrl2)

Example 4) GET:\$ENGINE\$::!ControllerNames: // Obtain the list of Controller name.

Example 5) GET:Ctrl1::!ItemNames:// Obtain the item name list of Controller (Ctrl1).

3.21.2. PUT command

The format of PUT command is as follows^{25 26}.

PUT:[<Controller-name>]:[<Item ID>]:<Item name>[!<Property name>]:<Type, Value>

Set a value to the property of a specified item. When <item ID> is entered, ID is set (put_ID) immediately before the acquisition.

When the controller name is omitted, the default controller name is used.

For available property names, refer to Table 3-10. If the property name is omitted, Value is used.

Example) PUT:Ctrl1::Item1:8, testok // Set "testok" to Item (Item1) of Controller (Ctrl1) by BSTR type.

3.22. CoAP server function

CoAP server function enables to exchange data with CaoSQL via the network using CoAP provider. Two or more server can be set.

To set CoAP server function, from the menu of CaoSQLConfig.exe, click [Action] -> [Option], and then select [CoAP] tab (Refer to 4.2.2.3). Specify the option character string of AddController of CoAP provider for

²⁵ (Note) You need to enter two colons (::) between the controller name and the item name.

²⁶ For details about how to write data types and values, refer to "2.2.5. Notation of data" of '[ORiN2 programming guide](#)'.

setting. For more details about CoAP provider, refer to [CoAP provider guide](#).

The following image shows the example of setting window of RAC Server function.

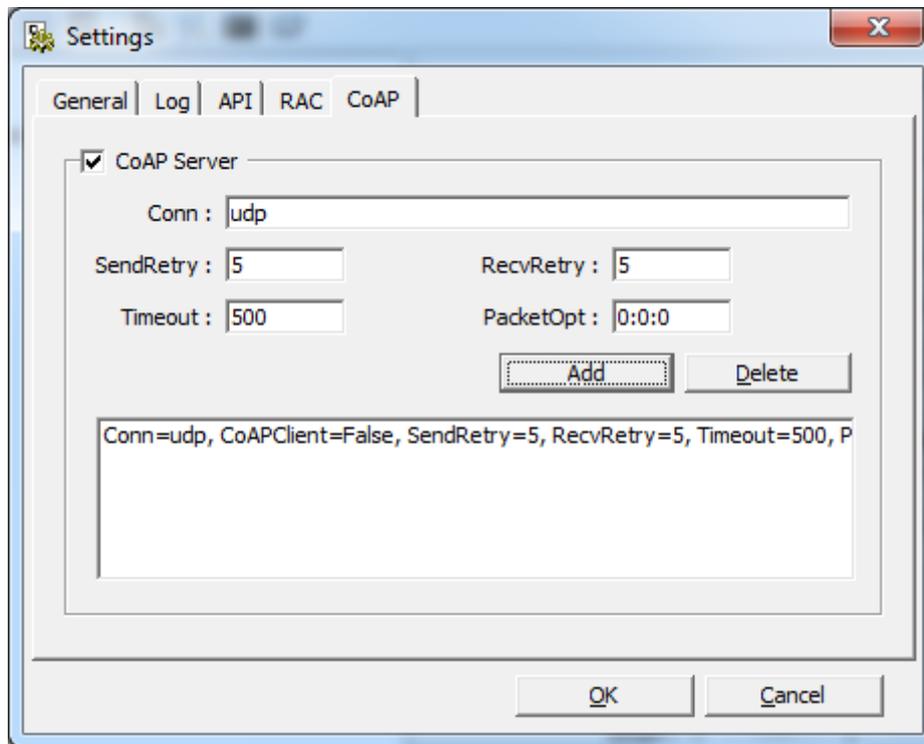


Figure 3-16 Example of setting CoAP server

The following introduces available CoAP commands.

3.22.1. GET command

To obtain a property value of the specified item of CaoSQL with CoAP protocol, send the following packet.

[Code]Get (1)

[Option] Value:URI path (11), Data:[<Controller name>/]<Item name>

When the controller name is omitted, the default controller name is used.

3.22.2. PUT command

To put a value to the specified CaoSQL item with CoAP, send the following packet.

[Code]Put (2)

[Option] Value:URI path (11), Data:[<Controller name>/]<Item name>

[Payload] VT_VARIANT

When the controller name is omitted, the default controller name is used.

3.23. Item dynamic registration/deletion function

Item dynamic registration/deletion function enables to register/delete items after CaoSQL start-up. Note that the dynamic deletion is available only to the item registered dynamically.

Item dynamic registration is executed by sending a VARIANT array that stores each data into controller's AddItem method. Data to be stored in each element of the array is predetermined as follows. (Assume that the minimum array subscript is 0.)

Table 3-11 Argument to AddItem method

Array element number	Data type	Content of data
0	VT_BSTR	Item name
1	VT_BOOL	Specify if CaoVariable is used
2	VT_BSTR	CaoVariable name
3	VT_BSTR	Option when CaoVariable is created
4	VT_I4	Parent class type of CaoVariable 0 Controller class 1 Task class 2 Robot class 3 Extension class 4 File class
5	VT_BSTR	Parent object name of CaoVariable
6	VT_BSTR	Detailed information
7	VT_I4	Read/Write attribute 0 Read/Write 1 Read 2 Write
8	VT_BSTR	Link item setting
9	VT_BOOL	Specify if Initialization is performed
10	VT_BSTR	Initialization value

11	VT_I4	Initialization type 0 VT_BOOL 1 VT_I1 2 VT_UI1 3 VT_I2 4 VT_UI2 5 VT_I4 6 VT_UI4 7 VT_R4 8 VT_R8 9 VT_BSTR 10 VT_DATE 11 VT_CY
12	VT_BOOL	Specify if VT_EMPTY is used
13	VT_BOOL	Specify if History specification is set
14	VT_BOOL	Specify if OnChange event is set
15	VT_I4	Specify if the value compulsive recognition is used. 0 None 1 Use at the time of reading 2 Use at the time of writing
16	VT_BOOL	Specify if the dead band setting is used
17	VT_R4	Minimum value of the dead band setting
18	VT_R4	Maximum value of the dead band setting
19	VT_BOOL	Specify if the valid range setting is used
20	VT_R4	Minimum value of the valid range setting
21	VT_R4	Maximum value of the valid range setting
22	VT_BOOL	Specify if Chattering setting is used
23	VT_I4	Chattering value
24	VT_BOOL	Specify if masking is used
25	VT_I4	Masking value
26	VT_BOOL	Specify if BCD conversion is used

27	VT_I4	<p>Operators for binary operation (the first operation)</p> <table> <tr><td>0</td><td>(NOP)</td><td>1</td><td>+</td></tr> <tr><td>2</td><td>-</td><td>3</td><td>*</td></tr> <tr><td>4</td><td>/</td><td>5</td><td>MOD</td></tr> <tr><td>6</td><td>AND</td><td>7</td><td>OR</td></tr> <tr><td>8</td><td>XOR</td><td>9</td><td><<</td></tr> <tr><td>10</td><td>>></td><td>11</td><td>=</td></tr> <tr><td>12</td><td><></td><td>13</td><td><=</td></tr> <tr><td>14</td><td><</td><td>15</td><td>>=</td></tr> <tr><td>16</td><td>></td><td>17</td><td>CAT</td></tr> <tr><td>18</td><td>IDIV</td><td>19</td><td>IMP</td></tr> <tr><td>20</td><td>POW</td><td>21</td><td>ABS</td></tr> <tr><td>22</td><td>FIX</td><td>23</td><td>INT</td></tr> <tr><td>24</td><td>NEG</td><td>25</td><td>NOT</td></tr> <tr><td>26</td><td>RAD2DEG</td><td>27</td><td>DEG2RAD</td></tr> </table>	0	(NOP)	1	+	2	-	3	*	4	/	5	MOD	6	AND	7	OR	8	XOR	9	<<	10	>>	11	=	12	<>	13	<=	14	<	15	>=	16	>	17	CAT	18	IDIV	19	IMP	20	POW	21	ABS	22	FIX	23	INT	24	NEG	25	NOT	26	RAD2DEG	27	DEG2RAD
0	(NOP)	1	+																																																							
2	-	3	*																																																							
4	/	5	MOD																																																							
6	AND	7	OR																																																							
8	XOR	9	<<																																																							
10	>>	11	=																																																							
12	<>	13	<=																																																							
14	<	15	>=																																																							
16	>	17	CAT																																																							
18	IDIV	19	IMP																																																							
20	POW	21	ABS																																																							
22	FIX	23	INT																																																							
24	NEG	25	NOT																																																							
26	RAD2DEG	27	DEG2RAD																																																							
28	VT_BSTR	Operation value for binary operation (the first operation)																																																								
29	VT_BOOL	Specify if the data type conversion is performed (at the time of reading).																																																								
30	VT_I4	<p>Data type setting of data type conversion (at the time of reading).</p> <table> <tr><td>0</td><td>VT_BOOL</td><td>1</td><td>VT_I1</td></tr> <tr><td>2</td><td>VT_UI1</td><td>3</td><td>VT_I2</td></tr> <tr><td>4</td><td>VT_UI2</td><td>5</td><td>VT_I4</td></tr> <tr><td>6</td><td>VT_UI4</td><td>7</td><td>VT_R4</td></tr> <tr><td>8</td><td>VT_R8</td><td>9</td><td>VT_BSTR</td></tr> <tr><td>10</td><td>VT_DATE</td><td>11</td><td>VT_CY</td></tr> <tr><td>12</td><td>VT_VARIANT</td><td></td><td></td></tr> </table>	0	VT_BOOL	1	VT_I1	2	VT_UI1	3	VT_I2	4	VT_UI2	5	VT_I4	6	VT_UI4	7	VT_R4	8	VT_R8	9	VT_BSTR	10	VT_DATE	11	VT_CY	12	VT_VARIANT																														
0	VT_BOOL	1	VT_I1																																																							
2	VT_UI1	3	VT_I2																																																							
4	VT_UI2	5	VT_I4																																																							
6	VT_UI4	7	VT_R4																																																							
8	VT_R8	9	VT_BSTR																																																							
10	VT_DATE	11	VT_CY																																																							
12	VT_VARIANT																																																									
31	VT_BOOL	Specify if the data type conversion is performed (at the time of writing).																																																								

32	VT_I4	Data type setting of data type conversion (at the time of writing). 0 VT_BOOL 1 VT_I1 2 VT_UI1 3 VT_I2 4 VT_UI2 5 VT_I4 6 VT_UI4 7 VT_R4 8 VT_R8 9 VT_BSTR 10 VT_DATE 11 VT_CY 12 VT_VARIANT
33	VT_BOOL	Specify if data comparison in array is performed.
34	VT_BOOL	Array element distribution link setting
35	VT_BOOL	Inactivation
36	VT_I4	Item type 0 General Item 1 Alias item 2 Array type item
37	VT_I4	Link destination property 0 Value 1 ID 2 Activation 3 LinkActivation
38	VT_I4	Operator for binary operation (the second operation)
39	VT_BSTR	Operation value for binary operation (the second operation)
40	VT_BSTR	Option string of CaoVariable's parent object.

3.24. Script execution function²⁷

Script execution function executes VB script. The script can be set and executed in each controller. At the controller startup, VB script configured by CaoSQLConfig (Figure 3-17) is read, and the subroutine Main in the script will be executed at the time of each sampling.

²⁷ This function doesn't operate in Microsoft Windows 7 (64bit version).

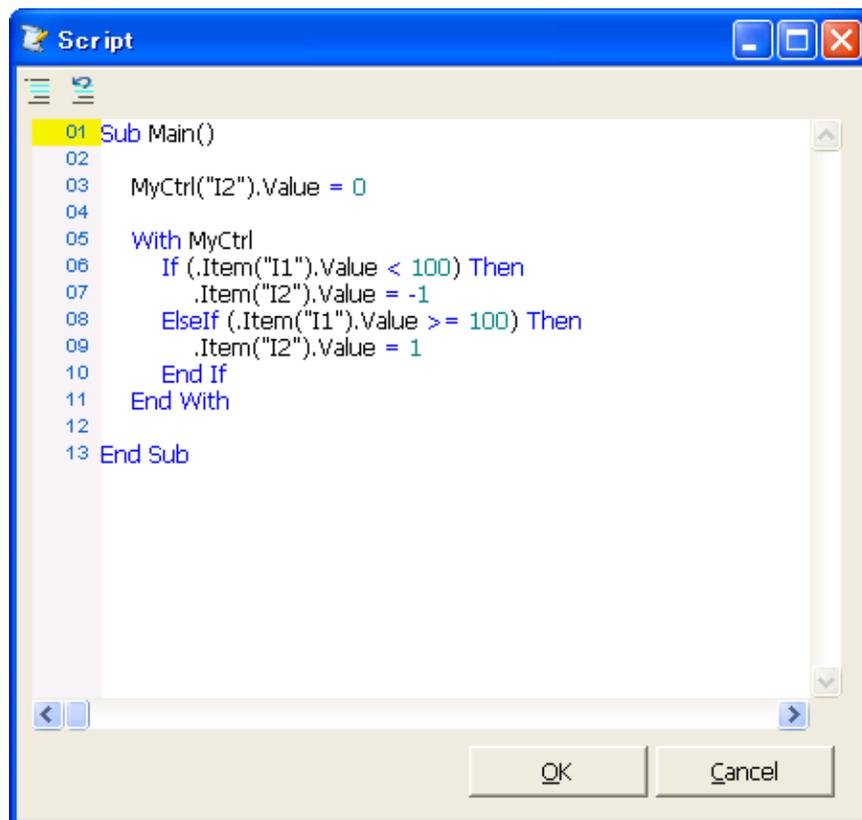


Figure 3-17 Script window

VB script completely complies with Microsoft VBScript specifications; however, the use of user interface-related functions (e.g; MsgBox, InputBox) and functions for object acquisition (e.g; CreateObject, GetObject) are prohibited because VB script works as a middleware of CaoSQL, as well as for the safety reason.

Instead of that, MyCtrl, which is an embedded object, is added. This is CaoSQLController object itself that configures the script. Therefore, you can easily set the calculation result in the item's value by using MyCtrl as follows.

Example) Copy the triple value of "I2" that is an own controller's item into "I1".

```
MyCtrl.Item("I1").Value = MyCtrl.item("I2").Value * 3
```

To set the calculation result to the different controller's item, use both the script execution function and the item link function. Concretely, set a value to the linked item. However, note that the value transmission delays by one cycle since the value is transmitted to the link destination in the next scanning.

A script is executed by calling subroutine Main at the end of sampling every time. Note that the sampling processing will stop if an infinite loop is written in Main. However, since threads on other controllers are not be stopped, these threads continue the processing in parallel. To control the start and stop of a script via API, use CSQ command (\$CSQ_SCRIPT_ENABLED\$) of Execute method. Scripts can be executed from the inside of a script by using MyCtrl object.

Example) Stop the execution of own controller's script

```
MyCtrl.Execute "$CSQ_SCRIPT_ENABLED$", False
```

If an error occurs within a script, the script stops. To resume the script, execute CSQ command (\$CSQ_SCRIPT_ENABLED\$) via API. To avoid the stop of the script, execute "On Error Resume Next" statement on the top of the script so that errors in the script is ignored.

You can obtain the details of error information with CSQ command (\$CSQ_SCRIPT_ERROR\$). Note that a startup error (or an error at the initialization), such as grammar error, cannot be obtained with this command. Startup error is output to the error log.

[Notes]

Script execution function can execute various operations, such as BCD conversion, binary operation. However, it requires huge overhead, so it is recommended not to use this function if you can execute the target operation without this function.

3.25. CSQ command

Execute method of CaoSQLEngine class and CaoSQLController support the following commands.

Table 3-12 CSQ command of CaoSQLEngine class

Command name	Argument	Description
(Not implemented)		

Table 3-13 CSQ command of CaoSQLController class

Command name	Argument	Description
\$CSQ_RESET_SCANTIME\$	None	Reset three of scan time measurement values (current value, minimum value, and maximum value). The current and maximum values are set to 0, and the minimum value is set to LONG_MAX.
\$CSQ_SCAN_ONETIME\$	None	Perform scanning (value acquisition) compulsorily, even if neither the controller nor the item are active. (Note) The control is not returned until the scanning of all items is completed because this command scans synchronously. Therefore, if this command is executed in the automatic sampling, processing

		might be blocked and takes long time to return.
\$CSQ_SCRIPT_ERRORS\$	None	Obtain the error information at the script execution by the following three array elements. (<Code>, <Line No>, <Description>) <Code> VT_I4 : Error code <Line No> VT_I4 : Line number <Description>VT_BSTR : Error details
\$CSQ_SCRIPT_ENABLED\$	VT_BOOL	Permit (True)/Prohibit(False) the execution of script at the time of sampling. If the default value is Permit (True), a script is loaded at the time of CaoSQL startup. If the default value is Prohibit (False), a script is loaded at the first sampling timing after this command is set to Permit (True).
Others		Redirect to Execute method of CaoController class for the corresponding provider.

3.26. The list of available data type information

The following table shows the data types that can be used in CaoSQL.

Table 3-14 Data Types available in CaoSQL

Function	Data type information
Event notification (item value change) (status change)	VT_EMPTY,VT_NULL,VT_BOOL,VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4,VT_R4,VT_R8,VT_CY,VT_DATE,VT_BSTR,VT_ERROR,VT_DISPATCH,VT_UNKNOWN
Array element extract function	VT_BOOL,VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4,VT_R4,VT_R8,VT_CY,VT_DATE,VT_BSTR,VT_VARIANT
Data masking function	VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4
BCD conversion function	VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4
Data type conversion function	VT_BOOL,VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4,VT_R4,VT_R8,VT_CY,VT_DATE,VT_BSTR,VT_VARIANT
Dead band setting function	VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4,VT_R4,VT_R8
Valid range setting function	VT_I1,VT_UI1,VT_I2,VT_UI2,VT_I4,VT_UI4,VT_R4,VT_R8

4. CaoSQLConfig

4.1. Outline

CaoSQL Configuration Manager (execution file name: CaoSQLConfig.exe., hereafter, CaoSQLConfig) is a tool that sets items of CaoSQL in a computer. Information set here is recorded in the registry and the CSQ file, and will be read when CaoSQL starts.

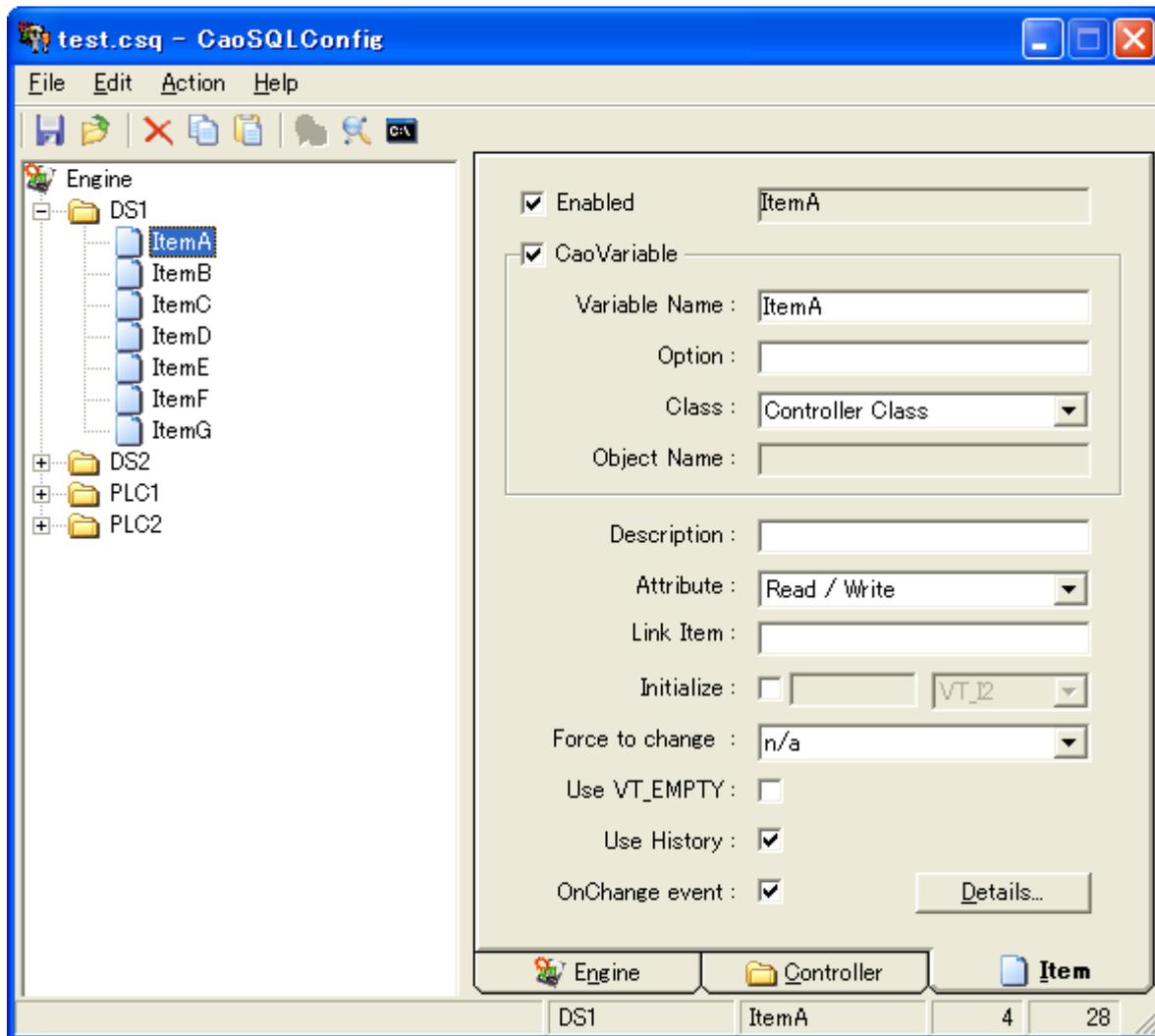


Figure 4-1 CaoSQLConfig window

4.2. Operation method

4.2.1. Tab input

There are three tabs on the right bottom of the screen. The tab displays information of the object corresponding to the selected node on the tree view. When a node of Engine is selected, the tab displays

Engine's information. When a node of Controller is selected, the tab displays Controller's information. When a node of Item is selected, the tab displays Item's information.

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

4.2.1.1. Engine tab

[Figure 4-2](#) shows the items of Engine tab. You can set information of Engine as you like. (A default name is entered automatically in the CaoSQL Engine name.)

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 main area with several sections. The 'CaoEngine' section has 'Location : localhost', 'Retry Count : 3' (with an 'Unlimited' checkbox), and 'Retry Interval (ms) : 3000'. The 'UPnP' section has 'Friendly Name : ORiN Device' and a 'UPnP...' button. The 'History' section has a checked checkbox, 'Table Type : History' (dropdown), and a 'Database...' button. The 'Default Controller : RC1' (dropdown) is at the bottom. At the very bottom, there are three tabs: 'Engine' (selected), 'Controller', and 'Item'.

Figure 4-2 Engine tab

The following items can be set in the Engine tab.

[Start computer name] – Location

Enter the start computer name or IP address of CAO.exe. When nothing is input, it starts with the local computer.

[Number of Retry, Unlimited retry] – RetryCount, Unlimited

If “Unlimited” checkbox is selected, a computer will retry opening CaoController when it fails to open. Please enter the retry count to “Number of retry” or select “Unlimited” checkbox.

[Retry interval] – Retry Interval (ms)

Specify how long it takes from the failure of the CaoController opening to the retry by milliseconds.

[History function] – History

History function is enabled if this check box is selected.

[Table type] – TableType

Set the method of writing the record in the table.

History data and the latest data can be distinguished by the item called “Flags” of the record.

- History

Write the history data in the database. Add the record to the database when the value of the item changes.

- Current Value

Write the latest data in the database. Overwrite the record of the latest data when the value of the item changes.

- History & Current Value

Execute “History” and “Current Value” at the same time.

[Default controller] – Default Controller

This controller name will be used if the controller name is not specified by the setting that specifies the controller,

[DataBase setting] – DataBase

History function (data logging) setting window appears.

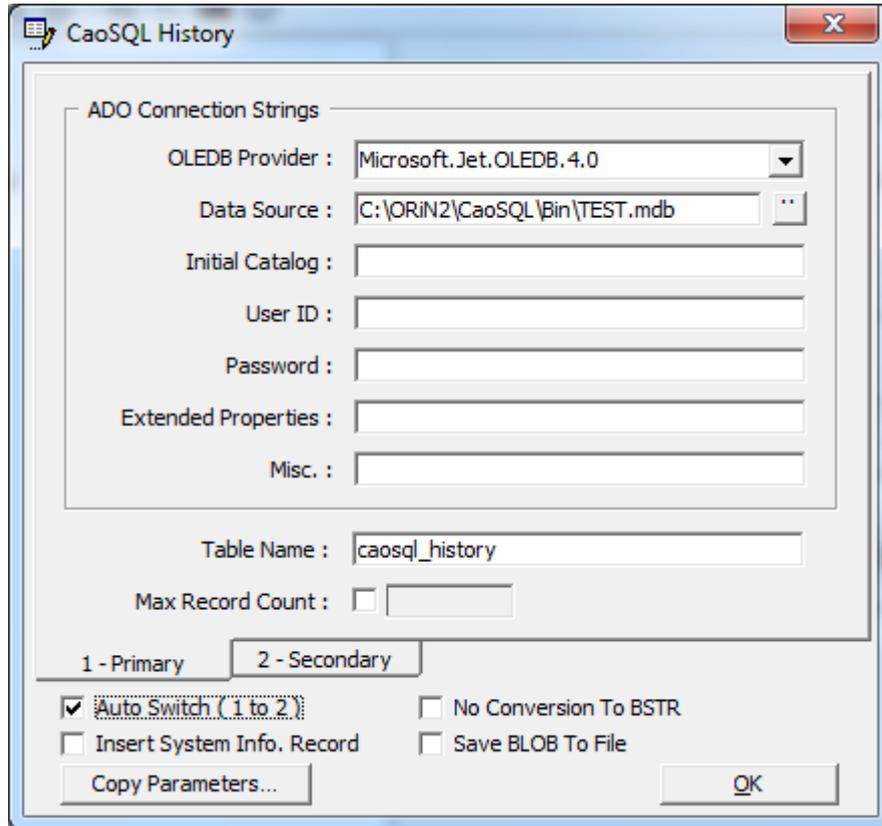
4.2.1.1.1. History function setting window

Figure 4-3 History setting window

[ADO connection string] – ADO Connection String

Select options that are used when the ADO connection opens.

To select other ADO connection string options, enter the followings in “Misc.” textbox.

<Item name to set>=<Content to set>;<Item to set>=<Content to set>;<Item to set>=<Content to set>]...

For details, refer to ADO reference.

[Table name] – Table Name

Enter the table name to be used in History function.

[The maximum record count] – Max Record Count

Specify the maximum record count for logging. Select the check box and then enter the maximum count

to the text box. The maximum record count is not monitored if the checkbox is not selected or if 0 is entered in the text box. Since this option requires huge overhead, it is recommended to turn it off unless it is really necessary.

[Auto switch] – Auto Switch

Enable/Disable the Secondary DB switch function.

To set the items of secondary DB, select “2 – Secondary” tab from the History function setting window.

[Not convert to the string data type] – No Conversion To BSTR

Specify whether or not to convert an item value to the string type at the recording.

[Save binary data to a file] – BLOB To File

Specify whether or not to save data to the file if item's value is binary data.

[Copy Parameters] – Copy Parameters

Clicking this button will display a pull-down list. (Figure 4-4)

The settings of ADO connection string, table names, and the maximum record count are copied to the secondary database from the primary database, or vice versa.

Also, you can copy ADO connection strings to clipboard by setting them as option string of AddController's argument of DB provider.

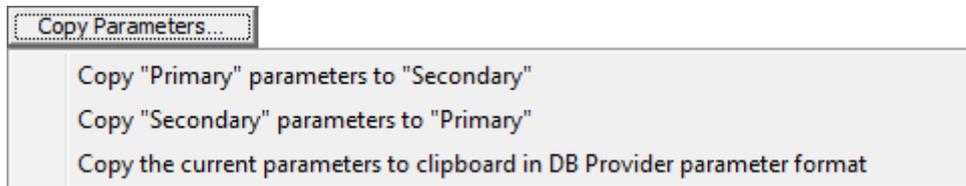


Figure 4-4 Copy parameters menu

4.2.1.2. Controller tab

Figure 4-5 shows the items of Controller tab.

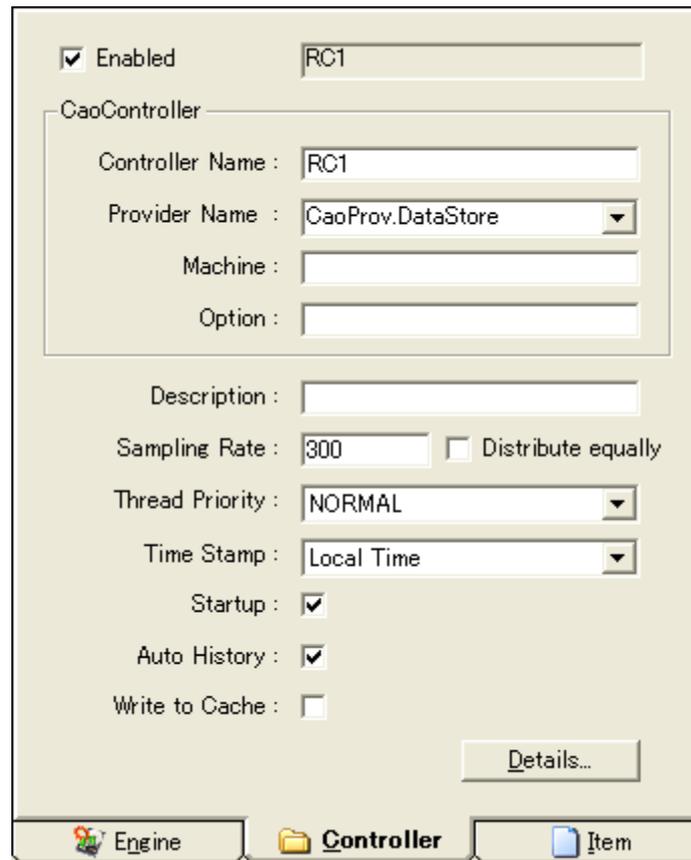


Figure 4-5 Controller tab

You can set information of Controller as you like; however, only “CaoSQLController name” is required to have a unique information within the registered information. To change “CaoSQLController name”, from the menu bar, click [Edit], select [Rename], and then enter a name. Controller name is not case-sensitive and a name that consists of digits only is not allowed. Symbols of \, \$, #, :, !, and / cannot be used.

The following Items can be set in the Controller tab.

[Enable/Disable setting of a controller] – Enabled

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

[Controller-name, Provider name, Computer name, Option]– Name, Provider Name, Machine, Option

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

[Sampling interval] – Sampling Rate

Specify the item polling interval by milliseconds. If the time to complete one polling is shorter than the specified sampling interval, the next polling will not be started until the specified sampling interval elapses²⁸.

[Even distribution] – Distribute equally

When Distribute equally check box is selected, CaoSQL delays the sampling by [*Sampling rate/Item count*] whenever the value of an item is obtained.

[Thread priority] – Thread Priority

Set the priority of the sampling thread for each controller. Sampling is performed in order from the highest priority²⁹. Set this option to NORMAL in normal use. The order of priority is as follows.

TIME CRITICAL > HIGHEST > ABOVE NORMAL > **NORMAL** > BELOW NORMAL > LOWEST > IDLE

TIME CRITICAL becomes top priority regardless of the process priority, and IDLE is set to the lowest priority. To change the process priority, click [**Action**]->[**Option**].

[Detailed information] – Description

Since this item can store character strings, it might be useful to write controller's information in it. If this item is set to invalid, it should be redirected to Help property of CAO provider's Controller object³⁰.

[Timestamp] – Time Stamp

Set the method to update the timestamp of the item.

- **n/a** : Timestamp is not updated
- **Device Time** : Update at the date of the variable object of CAO provider³¹.
- **Local Time** : Update at the local time and date.
- **System Time** : Update at the current system time and date expressed by Universal Coordinate Time (UTC)

[Start-up registration] – Startup

If this checkbox is selected, the sampling starts as soon as CaoSQL starts. If this checkbox is not selected, a thread is suspended-state when CaoSQL starts (Status is “deactive”), and the cache value is not updated. Writing is available while sampling is suspended³².

²⁸ (Reference) The recommended sampling rate is about 1/5 to 1/2 of the sampling interval requested by a client. For example, a client requests 100ms interval, you may set it between 20 and 50ms.

²⁹ The processing priority of OnChangeItem event thread and OnChangeState event thread on each controller are determined by the values specified here.

³⁰ Version 1.8.0. or higher.

³¹ Currently, Device Time is not implemented .

³² (Hint) When two or more CaoSQL is connected in the cascade when the variable of the CAO provider at the connection destination is a buffer type variable of the queue (First-In First-Out) and stack (Last In First Out) form and data is shared, the application in which the application to which data is read by using this option and data are written can be specified specifying it. (Please turn off the written application.)

However, if a thread reads a value while it is suspended-state, the cache value is updated at that timing and then the value is returned.

[Automatic recording of History] – Auto History

If this checkbox is not selected, although History settings of the controller's subordinate items are retained, History is not recorded. Also, the change of History setting on the Item tab is not allowed. When you record History explicitly by using Snapshot method, it is practical to remove all checks from the related items.

[Cache writing function] – Write to Cache

Turn on/off the cache writing function. If this checkbox is selected, an item's value is written in cache without waiting sampling. This is effective to execute get_Value of an item immediately after the execution of put_Value for the item.

[Script] – Script

Turn on/off the script execution function. If this checkbox is selected, a script is performed at the end of sampling. This function helps to set complicated conditions.

[Details] – Details

Clicking this button opens a Controller Details dialog window as shown below. This window consists of the following items : Watchdog, Trigger, and Auto Reconnection. The controller name currently set is displayed in the caption of the dialog.

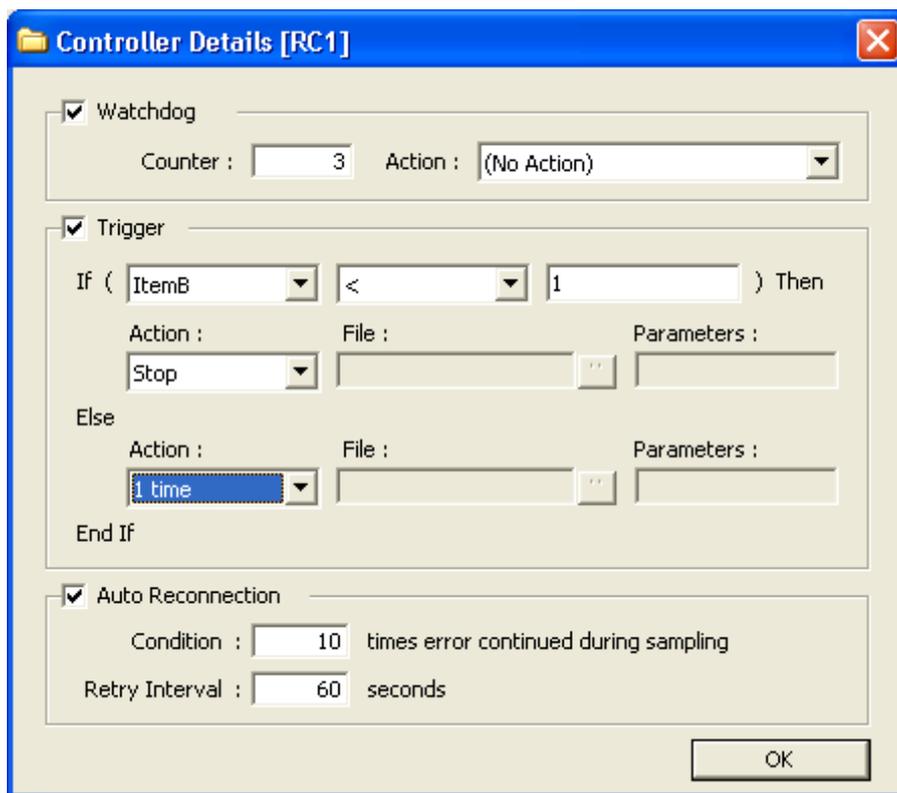


Figure 4-6 Setting of Controller Details

[Watchdog timer] – Watchdog

Perform the watchdog timer to set.

If set enabled, more than the specified number (the scope of **Counter**) of the sampling interval per 1 sampling will allow the controller status error and perform the proceeding selected with **Action**. If selected Stop with **Action**, sampling thread will be stopped.

[Note]

Evaluation of the sampling elapsed time is conducted per updating the item value; when the sampling interval has been passed at that time, its count will be increased by 1 as an exceeded time.

Thus, when the item number is smaller than the specified number, the watchdog timer cannot meet the condition. When updating the item value is fully slower than the sampling interval, the watchdog evaluation will be delayed.

[Trigger] – Trigger

If this checkbox is selected, trigger function becomes available.

- **STEP 1:**

For the conditional expression, select items to set a trigger. You can choose items of the controller

that the trigger is set.

- **STEP 2:**

Set the conditional expression and conditions for trigger execution. The conditions specified here are saved as character strings though, it will be converted along with the data type of the trigger-target item when the trigger is executed.

- **STEP 3:**

Set the action executed at the trigger execution. To perform any action at the opposite condition, enter the action in “Else” .

[Automatic reconnecting] – Auto Reconnection

If an error occurs by the number of times specified in **Condition**, the auto-reconnection will start once the period of time specified in **Retry Interval** has passed.

4.2.1.3. Item tab

[Figure 4-6](#) shows the item tab. Same as the controller tab, item names can be changed only from the menu bar. (From the menu bar, click [Edit] – [Rename]). Item name is not case-sensitive and a name that consists of digits only is not allowed. Symbols of \ , \$, # , : , ! , and / cannot be used.

Figure 4-7 Item tab

The following items can be set in the Item tab.

[Enable] – Enabled

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

[Deactivate (Stop auto-sampling)] - Inactivate

If this checkbox is selected, the reading or writing of items are not executed during the sampling interval of a controller.

[Enable CaoVariables] – CaoVariable

Set whether to bind the item to Variable class of CAO or not. If this option is set to invalid, the item will not be bound and will be treated as a local variable that is effective in CaoSQL only.

[Variable name] – Variable Name

This is an argument that is passed when a CaoVariable object is created (ICaoController::AddVariable). The input value is used as-is. This is case-sensitive.

[Option] – Option

Specify an option to add when you create a CaoVariable object.

[Class] – Class

Choose the type of Variable class.

[Parent object name] – Object Name

Enter the parent object name of a variable. If “Controller Class” is selected in Class selection (the above combo box), this text box becomes not available.

[System object option] – Object Option

Specify option strings at parent object creation. To display the selection window, click a [...] button next to [Class]. If "Controller Class" is selected in Class selection, the [...] button becomes invalid.

[Detailed information] – Description

Specify the detailed information on the item. Information set here is registered in CSQ file, so it is suitable for setting nonvolatile information. If this item is set to invalid, it will be redirected to Help property of CAO provider's Variable object³³.

[Initialization of value] – Initialize

Set an initial value to the value of an item. Although it is not quite practical for actual computer, it might be useful when you use DataStore provider or other similar provider a Controller.

[Link (Value transmission)] – Link Item

Set an item that is linked by the item link function. Specify the link item name by "Controller name \ Item name". If the link item name consists of “item name” only, it is recognized as an item on the same Controller. (Note that it is not a default controller name.). The consistency with the link destination item is not checked.

[Enable VT_EMPTY type] – Enable VT_EMPTY

Set whether to use VT_EMPTY as an effective data type or not. If this checkbox is not selected, VT_EMPTY is not regarded as an effective data. In this case, if the data type of the sampling target is VT_EMPTY, the change of the value is not detected because the sampling target is not recognized as an effective data. As a result, the succeeding processes, such as event notification or History processing, will not be performed.

[Automatic History recording] – Auto History

³³ Version 1.8.0 or higher.

When an item's attribute is "Read" or "Read/ Write", if this checkbox is selected, the value change is logged in the History database. However, this setting is not available when History function (data logging) is OFF-state or when the parent Controller's History function is OFF-state.

[Forcible value change] – Force to change

In the default setting of CaoSQL, OnChangeItem events and History recording are executed on the basis of the value change. If [1-Reading] is selected, the value (sampling result) is not compared with the previous one, and is considered to be changed always. As a result, CaoSQL executes OnChangeItem event or History recording at the time of each sampling.

Assume that CaoSQL performs sampling four times and the results are "1", "1", "2", "2". In the default setting, OnChangeItem event and OnChangeState event are issued two times and History recording is executed two times; at the time of the first result "1" is obtained, and at the time of value change from 1 to 2. However, if this option is enabled, each event occurs four times and History recording is executed four times³⁴. (History function should be enabled.)

If [2-Writing] is selected, when a value is updated from the external application, CaoSQL deems that the value is forcefully changed. This setting is available when [Write to Cache] is enabled or when [CaoVariable] is disabled.

[Enable/Disable setting for Event] – OnChange Event

Enable/Disable CaoSQL item event function.

³⁴ (Note) Note that when the History function is enabled, if the sampling interval is relatively short time, the large amount of data are recorded.

[Setting of Details] – Details

The screenshot shows the 'Item Details [DS - ItemA]' dialog box with the following settings:

- Data Filtering:**
 - Deadband: -95 - 10
 - Min-Max: -100 - 100
 - Chattering: 5
- Allowed Value:**
 - Min-Max-Step: 0 - 0 - 0
 - List: [Empty]
 - Default Value: [Empty]
 - Enable VT_EMPTY:
- Data Conversion:**
 - Masking: 64
 - BCD:
 - Calculation: * [DS\ItemD]
 - (NOP) [Empty]
 - Request Type (R): VT_R4
 - Request Type (W): VT_VARIANT
- Misc.:**
 - Data Type: VT_BOOL
 - BLOB: 0 [Empty]
 - EU Unit: [Empty]
 - Compare Arrays:
 - Extract Linking:
 - Linked Property: 0 : Value

Figure 4-8 Item Details setting dialog**[Dead band setting] – Dead-Band**

Enable/Disable the dead band and set its range. If the value changes within the range specified here, it is not regarded as the value change, therefore corresponding processing, such as OnChangeItem event or History processing, will not be performed.

[Valid range setting] – Min-Max

Enable/Disable the valid range and set its range. The change outside the range specified by minimum value and the maximum value is not recognized as the value change, therefore corresponding processing, such as OnChangeItem event or History processing, will not be performed.

[Chattering]- Chattering

If the value remains within the valid range for consecutive sampling counts specified here, the value is deemed as TRUE.

[Masking] – Masking

Mask the value to the item. Mask data is decimal notation.

[BCD conversion] – BCD

Perform BCD conversion for item's value. Values are converted when values are read and written.

[Binary operation] – Calculation

Perform binary operation for a value with another item. Values are converted only when values are read.

[Setting for the type of received value] – Request Type R/W

Convert the sampled data or the written data into the specified type.

[Data comparison in array] – Compare Arrays

When data is an array type, each element of the array is compared. When a value changes, OnChangeItem event notification processing and/or History processing is performed.

[Array element distribution link] – Extract Linking

Enable/Disable the array element distribution link function (Extract Linking option).

4.2.2. Menu

Select a basic operation in CaoSQLConfig 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.

4.2.2.1. File menu

Save and load the created data.

The file extension for [Open], [Save], and [Save As...] is "csq".

The file extension for [Import] and [Export] is "csx" is used.

[Create a file] – New

Create a new csq file.

[Read from a file] – Open...

Read the set up information from the specified csq file³⁵.

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

[Save a File] – Save

Overwrite the contents of current setting in the csq file.

[Save a file as a different name] – Save As...

Save the contents of current setting in another csq file.

³⁵ 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

[Importing] – Import...

Import a node from the csx file to the currently selected node.

[Export] – Export...

Export the selected node to the csx file.

[Import from CSV file] - Import from CSV file...

Import a node from the csv file to the currently selected node.

[Export to CSV file] -Export to CSV file...

Export the selected node to the csv file.

[Exit] – Exit

Close the program of CaoSQLConfig.

4.2.2.2. Edit menu

Edit items, such as adding and deleting items.

[Add a new Controller] – Add Controller

Add a new controller that is read by CaoSQL. A controller which name is the same as an existing controller name cannot be added (The system is not case-sensitive.). A controller with valid name will be added to the tree. Enter detailed information of the controller on the Controller tab at the right side pane.

[Add a new Item] – Add Item

Add an item to the node selected by the tree view. An item which name is the same as an existing item name in the target Controller cannot be added (The system is not case-sensitive.). General items cannot be added to an alias controller. An item with valid name will be added to the item tree. Enter detailed information of the item on Item tab at the right side pane.

[Add an array type Item] – Add Array Item

Add an array type item to the node selected by the tree view. Just like “Add Item”, an item which name is the same as an existing item name in the target Controller cannot be added (The system is not case-sensitive.). Array Items cannot be added to an alias controller.

[Add an Alias Controller] – Add Alias Controller

Add an alias controller. An alias controller with the same name as the existing controller and alias

controller cannot be added.

[Add a Group] – Add Group

Add a group. Group is used to categorize items. It is also used as a label for categorized items.

[Name change] – Rename

Change the name of Controller/Item selected on the tree view. An already existing Controller/Item name cannot be used (The system is not case-sensitive.).

[Delete] – Delete

Delete a node selected on the tree view. If a controller is deleted, all subordinate items of the controller are also deleted. Selecting a node on the tree view and pressing Del key of the keyboard also deletes the selected node.

[Copy a Node] – Copy

Copy a node selected on the tree. The registered items under a controller are also copied. However, trigger information is not copied. The clone of the copied information is added to the node by [Paste] operation.

[Paste] – Paste

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

[Sorting] – Sort

Sort the nodes of the selected level in ascending order.

[Copy of DDE character string] – Copy DDE String

Copy the DDE character string of the item selected by the tree view onto the clipboard.

4.2.2.3. Action menu

Action menu offers execution of CaoSQL.exe and settings for CaoSQL.exe execution. The setting is saved in the registry.

[Option setting] – Settings

An environmental setting dialog of CaoSQL is displayed.

[General tab] – General

This tab provides CaoSQL general settings, such as the process priority and Locale ID..

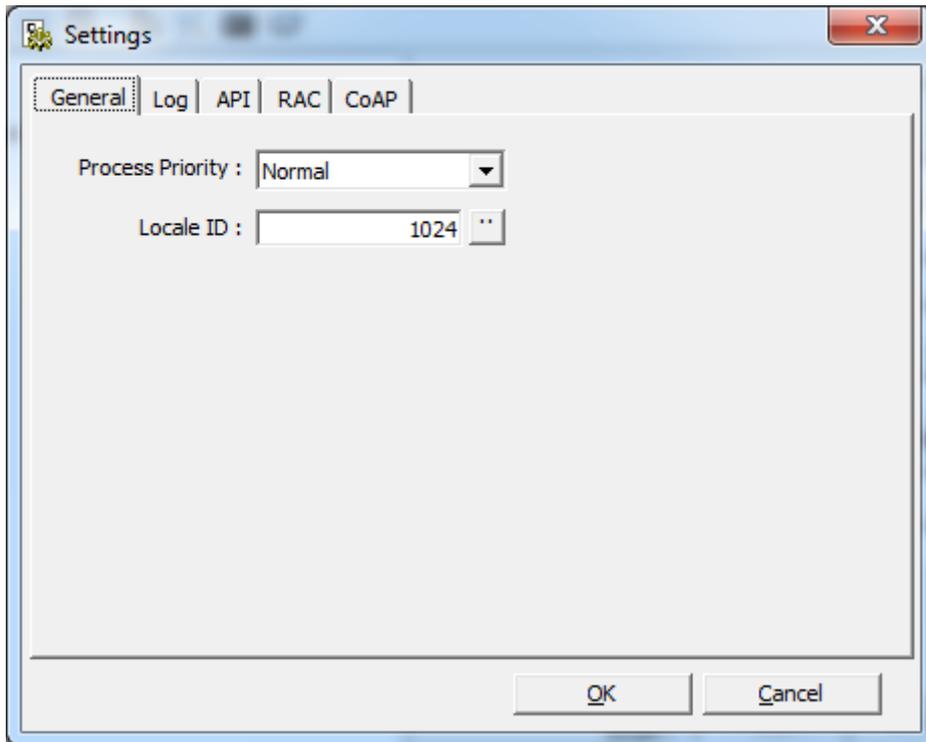


Figure 4-9 CaoSQL environmental setting (General)

[Process priority] – Process Priority

Set the process priority of CaoSQL. The following shows the order of priority.

REAL TIME > HIGH > **NORMAL** > IDEL

[Locale ID] – LocaleID

Set language ID to use.

[Log setting tab] – Log

This tab provides information settings of CaoSQL log output.

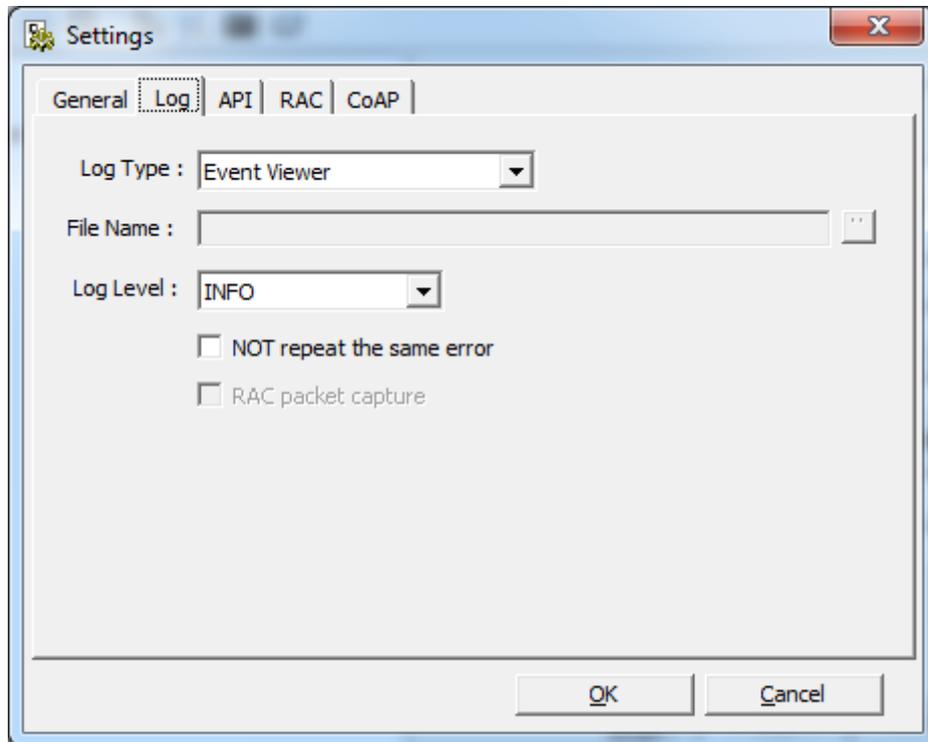


Figure 4-10 CaoSQL environmental setting (Log)

[Log type] – Log Type

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

The following output types are available.

Table 4-1 Log type

Output destination	Description
Console	Output to the console.
Message Box	Output to the message box (when service starts.).
Event Viewer	Output to the event viewer (when service starts.).
Debug Viewer	Output debug information.
Text File	Output to the specified text file.

[File name] – File Name

Specify the file path that outputs the log when the type of log is Text file.

[Log output level] – Log Level

Select an output log level from the following five levels. FATAL is the highest and DEBUG is the lowest severity level. The default setting is “INFO”.

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

[Same error output control] – NOT repeat the same error.

Set the output control for the same error output. If this checkbox is selected, an error message is output only once no matter how many times the same error occurs.

[API setting tab] – API

Set an API used in CaoSQL, such as DDE and JNI.

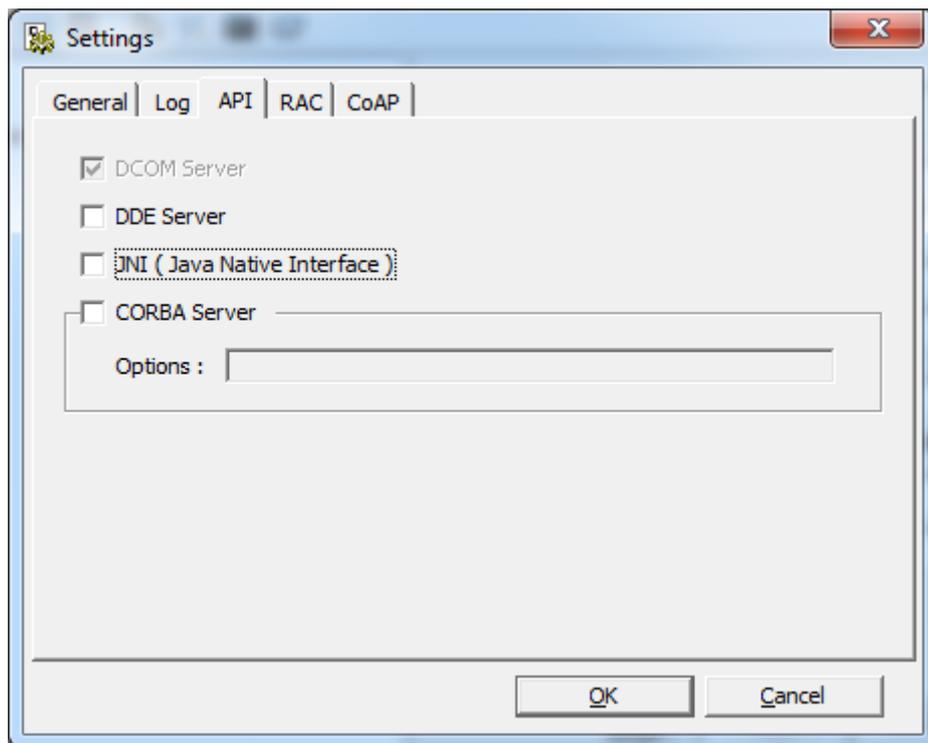


Figure 4-11 CaoSQL environment setting (API)

[DCOM server function] – DCOM Server

Enable/Disable the DCOM server function (always “Enable”).

[DDE server function] – DDE Server

Enable/Disable the DDE server function.

[JNI function] – JNI(Java Native Interface)³⁶

Enable/Disable the access through JNI.

[CORBA server function] – CORBA Server³⁷

Enable/Disable the CORBA server function. Use the character string input to Option as an argument.

[RAC function setting tab] – RAC

Set the RAC server to exchange data with CaoSQL by using RAC. Please refer to "2.2.1 AddController" of "[RAC provider guide](#)" for details.

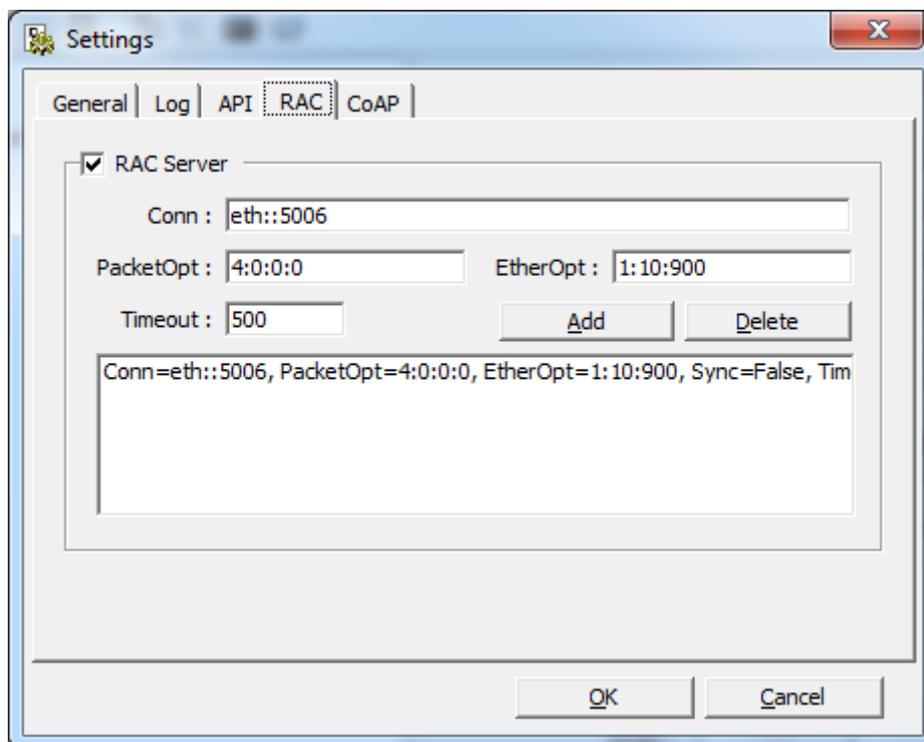


Figure 4-12 CaoSQL environmental setting (RAC)

[Enable/Disable the RAC server function] – RAC Server

Enable/Disable the RAC server function.

[Connection parameter of RAC server] – Conn, PacketOpt, EtherOpt, Sync, Timeout

³⁶ Currently, JNI function can be used regardless of this setting..

³⁷ Currently, this function is not implemented.

Set arguments to the AddController method of the RAC provider.

[Add a RAC server] – Add

Create an option argument for AddController method of the RAC provider by the specified parameter.

[Delete a RAC server] – Delete

Delete a currently selected character string that has been added by [Add a RAC server].

[CoAP function setting tab] – CoAP

Set the CoAP server to exchange data with CaoSQL data by using CoAP. Please refer to "2.2.1 AddController" of "[CoAP provider guide](#)" for details.

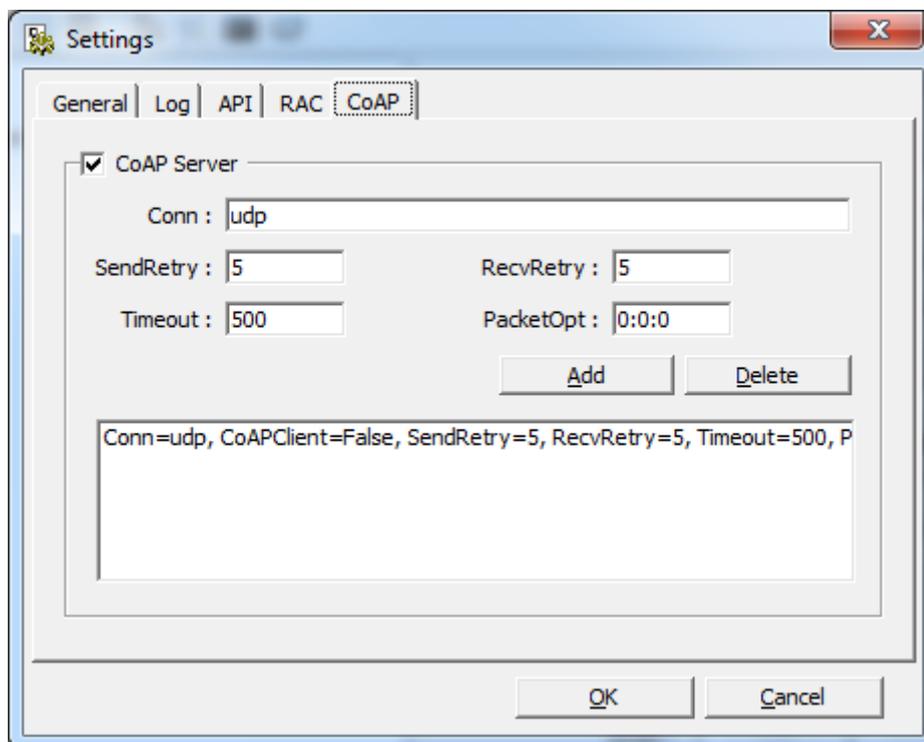


Figure 4-13 CaoSQL environment setting (CoAP)

[Enable/Disable CoAP server function] – CoAP Server

Enable/Disable the CoAP server function.

[Connection parameter of CoAP server] – Conn, SendRetry, RecvRetry, Timeout, PacketOpt

Set arguments to the AddController method of the CoAP provider.

[Add a CoAP server] – Add

Create an option argument for AddController method of the CoAP provider by the specified parameter.

[Delete a CoAP server] – Delete

Delete a currently selected character string that has been added by [Add a CoAP server].

[Start of service] – Service Start

Start the CaoSQL service. To update the csq file service, you need to restart CaoSQL service because the csq file information is read at the startup of CaoSQL. This is not available if CaoSQL.exe is not registered as a service or if the service is already executed.

[Stop of service] – Service Stop

Stop the CaoSQL service. This is not available if CaoSQL.exe is not registered as a service or if the service is already stopped.

[Restart of service] – Service Restart

Restart the CaoSQL service. This is not available if CaoSQL.exe is not registered as a service.

[Execution of CaoSQLTester]

Start the CaoSQLTester tool. Internally, this calls Shell function as it is.

[Execution of CaoSQLCmd]

Display the dialog for input of Shell command. Because the path of CaoSQLCmd.exe is displayed as a default, user can execute it by adding the command of CaoSQLCmd (see [Figure 4-13](#)). Internally, it calls Shell function as it is.

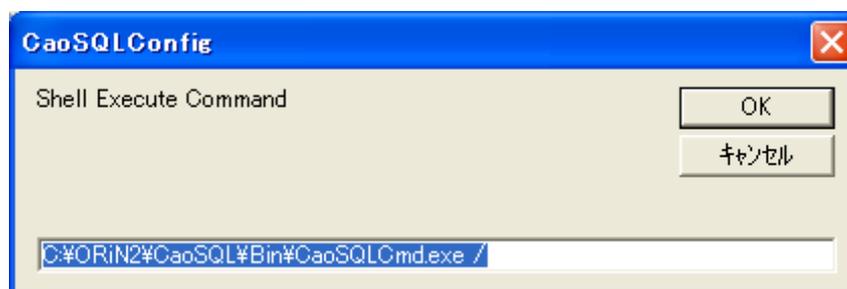


Figure 4-14 Execution dialog of Shell command

[Execution of CaoSQLLauncher]

Start the CaoSQLLauncher tool. Internally, it calls Shell function as it is.

4.2.2.4. Help menu

Help menu contains CaoSQL Help and the license information.

[Version information]

Display version information.

4.3. DENSO Robot mode

CaoSQLConfig can be started with a DENSO Robot mode that is dedicated mode for “DENSO NetwoRC Provider” or “DENSO RC8 Provider”.

To start CaoSQLConfig with one of these modes, add an option string “denso” to the startup file.

```
<ORiN2 install directory>\CaoSQL\Bin\CaoSQLConfig denso
```

4.3.1. Add controller (DENSO Robot mode)

Controller is added as the same way as the normal CaoSQLConfig mode.

In DENSO Robot mode, a wizard type dialog is displayed when a controller is added.

Select a provider that you want to add, enter a controller name, and then click [Next] button.

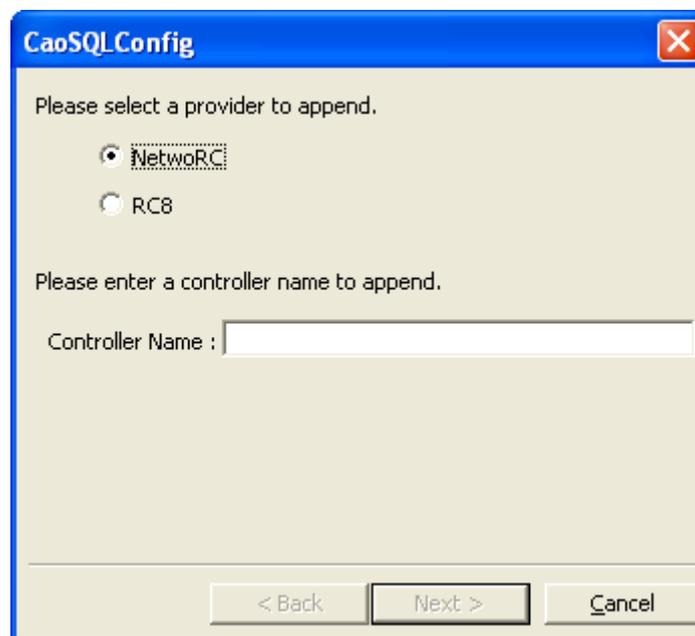


Figure 4-15 Adding controller in DENSO Robot mode

Next, set parameters of the connection destination controller. Enter an IP address for Ethernet connection, or set port number and baud rate for RS232C connection.([Figure 4-15](#))

If RC8 provider is selected, enter IP address only. ([Figure 4-16](#))



The screenshot shows the 'CaoSQL Config' dialog box with the 'Ethernet' radio button selected. The 'IP Address' field contains '192 . 168 . 0 . 1'. The 'RS232C' radio button is unselected. Below it, several fields are visible: 'Port No.' with a dropdown menu showing 'COM1', 'Baudrate' with a dropdown menu showing '9600bps', 'Parity' with a dropdown menu showing 'N - None', 'Data Length' with a dropdown menu showing '8', and 'Stop Bit' with a dropdown menu showing '1'. At the bottom, there are three buttons: '< Back', 'OK', and 'Cancel'.

Figure 4-16 Setting of parameter for communication (NetwoRC Provider)

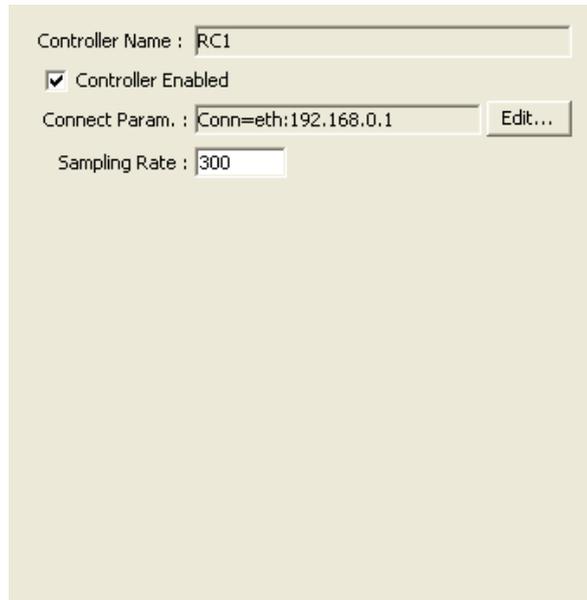


The screenshot shows the 'CaoSQL Config' dialog box with the 'Ethernet' radio button selected. The 'IP Address' field contains '192 . 168 . 0 . 1'. The 'RS232C' radio button is unselected. The rest of the dialog box is empty. At the bottom, there are three buttons: '< Back', 'OK', and 'Cancel'.

Figure 4-17 Setting of parameter for communication (RC8 Provider)

Click [OK] button to add a controller. If a controller with same name already exists or the name includes forbidden character, the addition will fail. (4.2.1.2)

If a controller is successfully added, the following dialog will be displayed. To change the connection destination, click [Edit] button. The second page of the wizard window is displayed ([Figure 4-17](#), [Figure 4-18](#)).



Controller Name : RC1

Controller Enabled

Connect Param. : Conn=eth:192.168.0.1 Edit...

Sampling Rate : 300

Figure 4-18 Setting controller name (NetwoRC Provider)



Controller Name : RC1

Controller Enabled

Connect Param. : Server=192.168.0.1 Edit...

Sampling Rate : 300

Figure 4-19 Setting controller name (RC8 Provider)

To avoid accessing the controller from CaoSQL, remove a check from the checkbox. You can set the sampling interval of this controller performed by CaoSQL by milliseconds.

4.3.2. Add item (DENSO Robot mode)

An item is added as the same way as the normal CaoSQLConfig mode.

In DENSO Robot mode, a wizard dialog is also displayed when an item is added.

Enter an item name. ([Figure 4-19](#))



Figure 4-20 Adding item

Next, select a variable. Select a variable class type. For “DENSO NetwoRC” ,“DENSO RC8”, select a class from “Controller”, “Task”, “Robot”, “Extension” or “File”. ([Figure 4-20](#), [Figure 4-21](#))

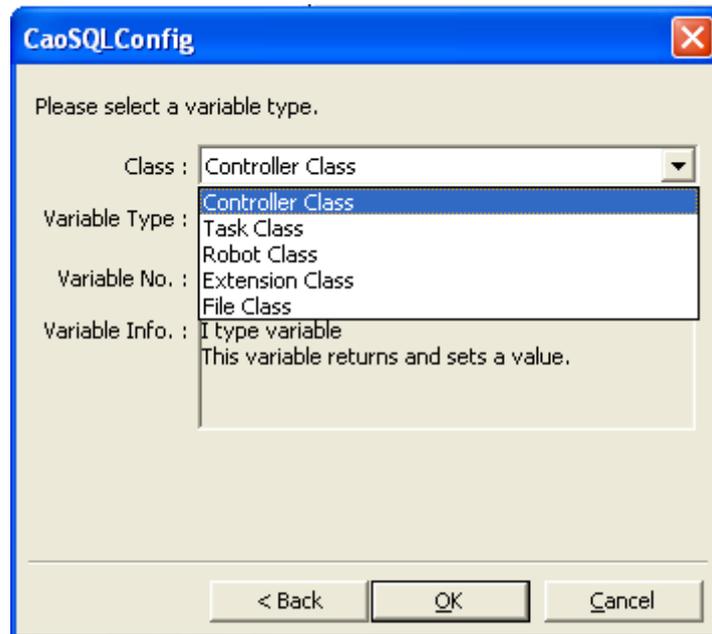


Figure 4-21 Select a class (NetwoRC Provider)

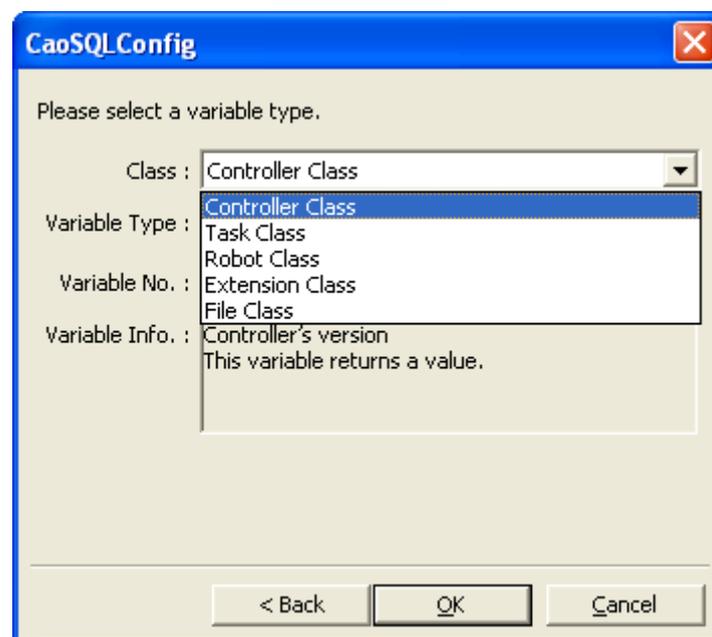


Figure 4-22 Select a class (RC8 Provider)

Select a class to which target variable belongs, because available variables differ depending on the type of the class. Select a variable suitable for the purpose of usage from the class. (Figure shows Robot class example)

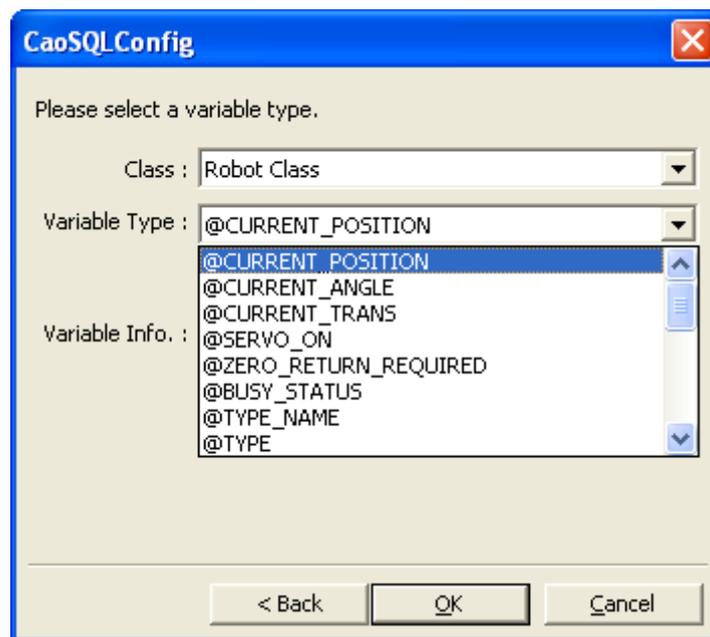


Figure 4-23 Select the type of variable (NetwoRC Provider)

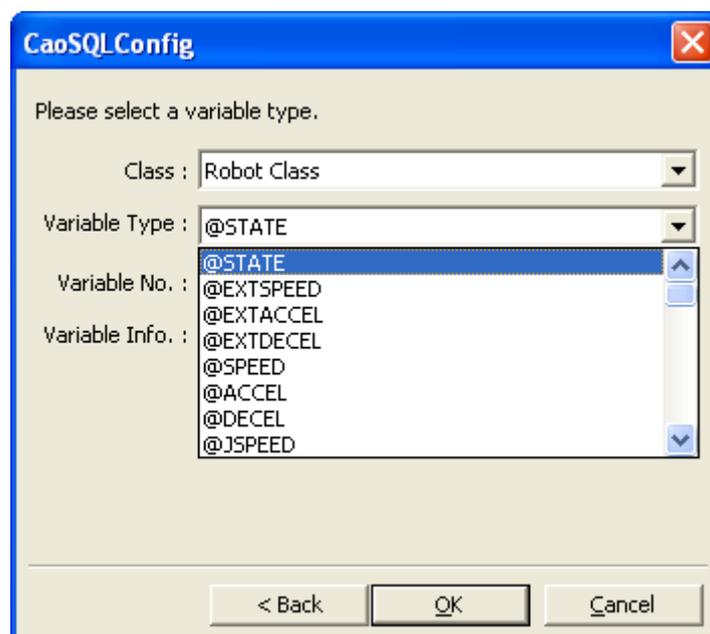


Figure 4-24 Select the type of variable(RC8 Provider)

For a variable that requires a variable number, enter a number in [Variable No]. [Variable Info] describes each variable. Item is properly added when [OK] is clicked, as the following image shows. To change the type of the variable, click [Edit...] button. The dialog which is displayed when the item is added will be displayed again.

Item Name : NetwoRC

Variable Enabled

Class : Controller Class Edit...

Variable Name : I11

Object Name :

Variable Info. : I type variable
This variable returns and sets a value.

Figure 4-25 Item tag in DENSO Robot mode

To avoid accessing the item from CaoSQL, remove a check from the [Variable Enabled] checkbox.

4.3.3. Loading configuration file for normal mode CaoSQL

Setting files saved in Normal mode can be read by the DENSO robot mode.

However, when other files than “NetwoRC Provider” or “RC8 Provider” are set to the setting file, controller tab and item tab are not displayed. In that case, the following message is displayed.

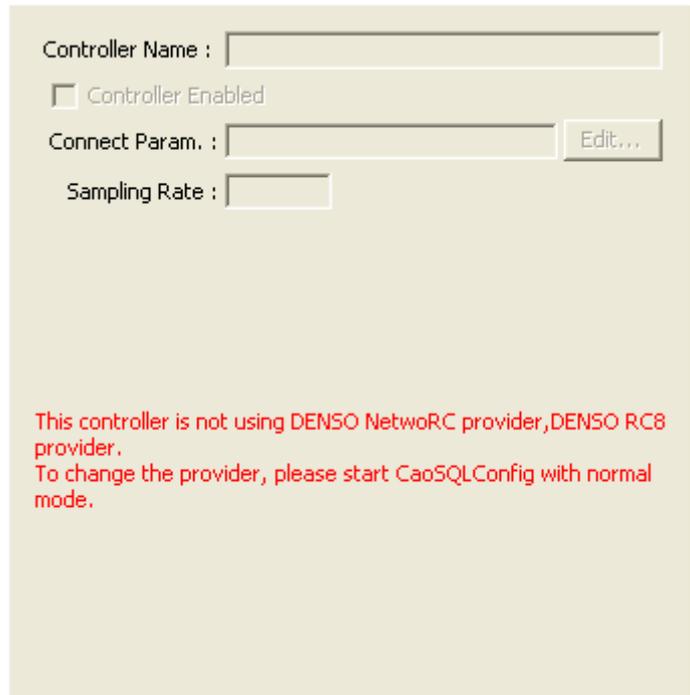


Figure 4-26 Controller tab disabled because of the provider difference

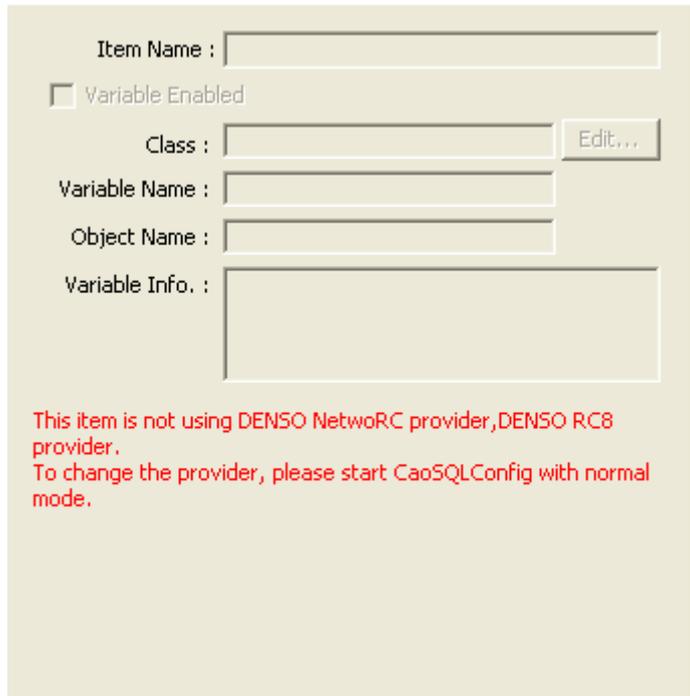


Figure 4-27 Variable tab disabled because of the provider difference

[Notes]

If a variable name is not available in “NetwoRC Provider” or “RC8 Provider”, the program fails to access with CaoSQL. In this case, the following warning message is displayed.



Figure 4-28 The warning message of using an illegal variable-name

[Notes]

If the variable number is invalid, the following message is displayed.

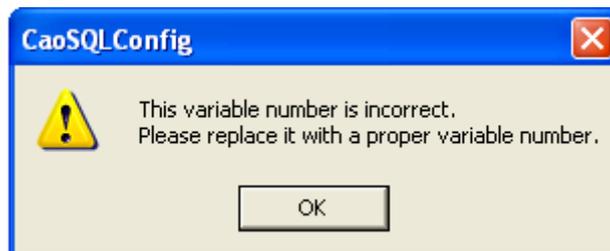


Figure 4-29 The warning message of using an illegal variable-number

4.4. Registry structure information

The registry structure set by CaoSQLConfig is shown below.

In CaoSQLConfig, items written in bold letters can be changed.

Key to registered registry: HKEY_CLASSES_ROOT\Software\CaoSQL

[]:Key Name

<>:Entry Name

{ }:Numeric Character

[CaoSQL]

---<App.EXENAME>	String	EXE name of CaoSQLConfig tool
---<App.Path>	String	Path name of CaoSQLConfig tool

---<CORBAEnabled>	DWORD	Effective/invalid switch of CORBA server
---<CORBAOption>	String	Start option of CORBA server
---<DataPath>	String	Path name of csq file
---<DCOMEnabled>	DWORD	Effective/invalid switch of DCOM server (always TRUE)
---<DDEEnabled>	DWORD	Effective/invalid switch of DDE server
---<FileName>	String	Path name when log type is file
---<JNIEnabled>	DWORD	Effective/invalid switch of JNI
---<LCID>	DWORD	Locale ID
---<LogLevel>	DWORD	Log level setting
---<LogType>	DWORD	Log type setting
---<NotRepeatError>	DWORD	Repetition setting of error log
---<ProcessPriority>	DWORD	CaoSQL process priority
---<RACCapture>	DWORD	Effective/invalid switch of RAC Capture
---<RACS {sequential number} >	String	Option character string of RAC server
---<RACServerCount>	DWORD	Number of registration of RAC servers
---<RACServerEnable>	DWORD	Effective/invalid switch of RAC server

4.5. Tips for Setting

This section explains useful settings for CaoSQL.

4.5.1. Decomposition of array element

One element of an array can be set as one CaoSQLItem object by using Array element extract function (Reference to 3.10), Item link function (Reference to 3.17), and CaoVariable disablement when the array is stored in the value of a certain item.

This setting has the following advantages:

- Efficient operation because sampling is done only once.
- The simultaneity of the value of each element is secured.
- The data processing can be set on each element (mask processing etc.).
- Each element of array can be recorded in History, although the History function does not support array.

The following shows practical examples by using DataStore provider.

Example) Obtain values of element number 0 and element number 1 of Item (array)

- (1) Create a Controller name "CtrlA". Set "CaoProv.DataStore" as a provider name. Select a checkbox of "Write to Cache".



Figure 4-30 Cache writing function setting window

- (2) Create an Item name "ItemA". Set "CaoVariable" to Enable. Enter "Item(0),ItemA(1)" in the Link Item textbox..

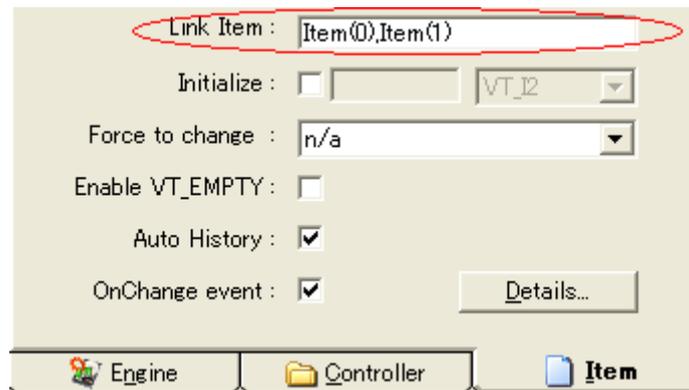


Figure 4-31 Link item setting window

- (3) Create an Item name "ItemA(0)" and "ItemA(1)". For both items, set CaoVariable to Disable.



Figure 4-32 CaoVariable setting window

4.5.2. Decomposition of bit element

With a data masking function (refer to 3.11) and Item link function (refer to 3.17), one bit of an item value can be set as an CaoSQLItem object.

This setting has the following advantages:

- Efficient operation because sampling is done only once.
- The simultaneity of the value of each element is secured.
- The data processing can be set on each element (mask processing etc.).
- Data can be recorded in History by bit-unit³⁸.

³⁸ This operation is not recommended since the overload may occur.

The following shows practical examples.

Example) Obtain the first and the second bits of the item value of the DataStore provider.

- (1) Create a Controller name "CtrlA". Set "CaoProv.DataStore" to the provider name.
- (2) Create an Item name "ItemB". Select the checkbox of CaoVariable
- (3) Create an Item name "ItemB1". From the Item Details window, select the checkbox of Masking, and enter 1 in the [Masking]textbox.
- (4) Create an Item name "ItemB2". From the Item Details window, select the checkbox of Masking, and enter 2 in the [Masking] textbox.

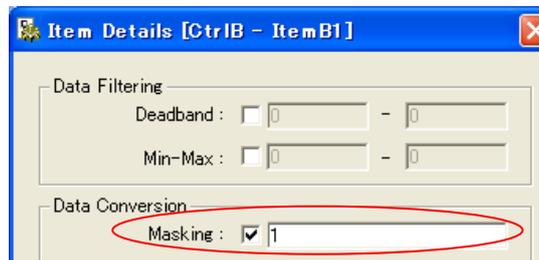


Figure 4-33 Mask value setting window

4.5.3. Polynomial operation

If 'Item link function' is used together with 'Binary operation function' (refer to 3.13), a complicated operation can be decomposed into simple binary operations.

The following shows practical examples.

Example) Setting for the processing "A<-A+B*C-D³⁹".

- (1) Create item C and item D.
- (2) Create a temporary item T1. With an Details window of Item, set "-D" to Calculation.
- (3) Create item B, set a temporary item T1 to the link item. With an Details window of Item, set "*C" to Calculation.
- (4) Create an item A. With an Details window of Item, set "+T1" to Calculation.

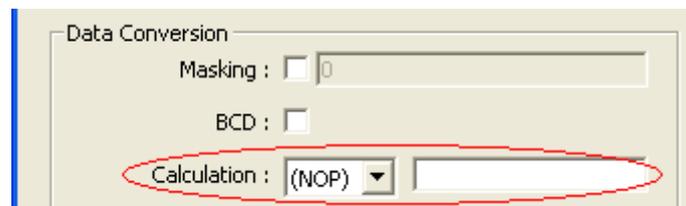


Figure 4-34 Binary operation setting window

³⁹ You may use the DataStore provider of CAO (provider for the data sharing) for such a usage. Binary operation function that is available from Ver.1.8.0 or higher helps to calculate complex operations.

- (5) Start CaoSQLTester. Once values are set to item A, B, C, and D, the calculation result of $A+B*C-D$ is set to the value of item A.

4.5.4. Conditional Item Link

By combining Trigger function (refer to 3.3) and Item trigger function (refer to 3.17), data can be transmitted to another controller's item only when certain items satisfy conditions.

For instance, to achieve the structure described in [Figure 4-35](#), the setting process is as follows.

- (1) Create a controller name "CtrlA" and "CtrlB". Set a certain device Provider to "CtrlA". Set "CaoProv.DataStore" to "CtrlB".
- (2) Create "ItemA" and "ItemB" under the controller "CtrlA". For each Items in Variable Name, enter a device's variable. Create "ItemC" under the Controller "CtrlB", and then set "Item" in "Variable Name".
- (3) From the Details window of Controller "CtrlA", select Trigger checkbox. From the top, select "ItemA" for the target item of condition, select "=" from the operator selection, enter "1" for the value text box, select "Start" from [Action] (displayed above), select "Stop" from [Action] (displayed below).

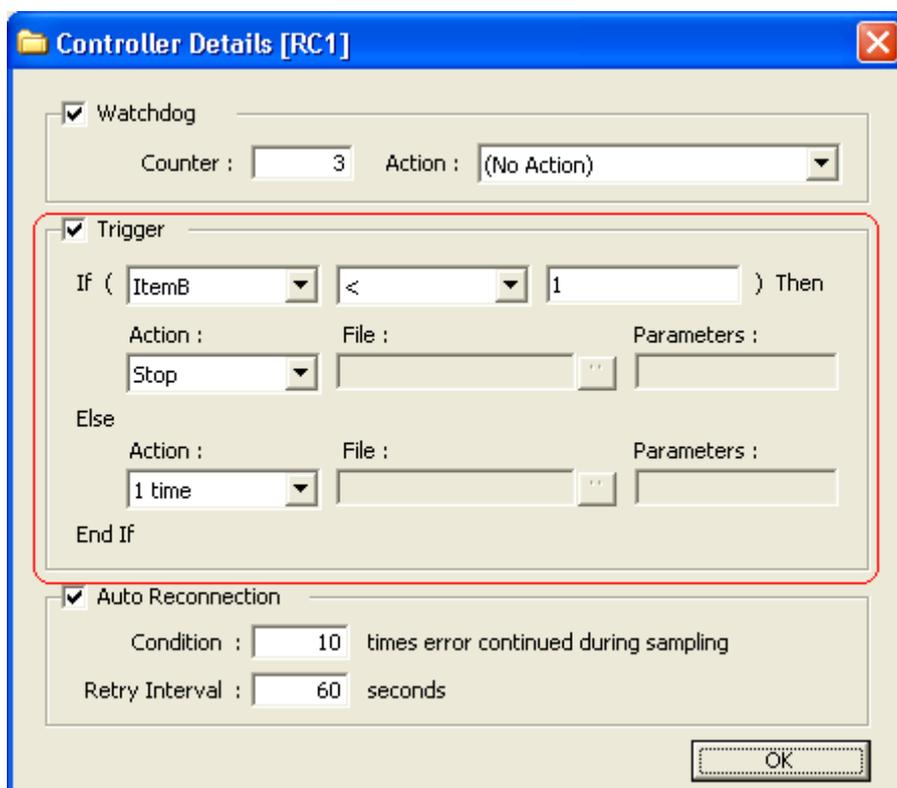


Figure 4-35 Trigger setting window

- (4) For "Link Item" textbox under "ItemB" of "CtrlA", enter "CtrlB\ItemC".

This completes the necessary settings. The program is executed as follows.

When “ItemA”, which is an item of “CtrlA”, is overwritten by an external device, once the trigger condition of “CtrlA” is satisfied, all items registered in “CtrlA” start sampling (For information of the trigger operation, refer to 3.3). “ItemB”, which is another item of “CtrlA”, also starts sampling. Once the value of “ItemB” changes, the value is transmitted to “ItemC” that is an item of “CtrlB”.

If a value other than the trigger condition is set, the sampling of all “CtrlA” items except for the trigger-target item will stop. Once the trigger condition is satisfied, value transmission will resume.

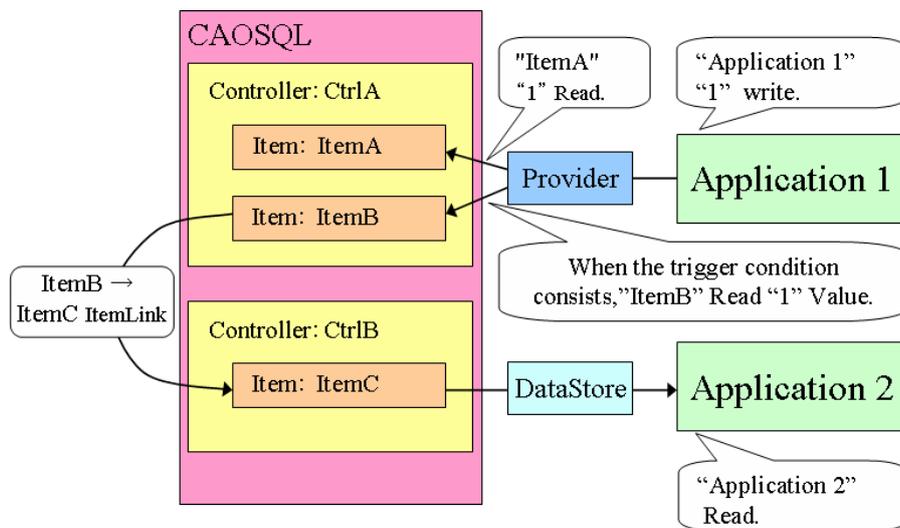


Figure 4-36 Processing flow of item transmission on trigger condition

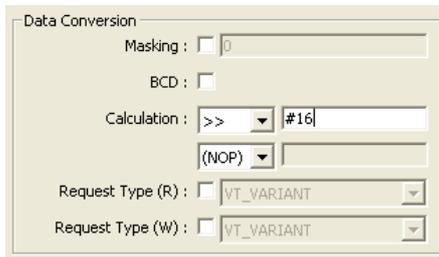
4.5.5. Decomposition of one VT_I4 type data into two VT_I2 type data

This section explains how to break up one VT_I4 (long type) into two VT_I2 (short type).

The following is an example of the decomposition.

Example: Break up “1048575”(&HFFFFFF) of VT_I4 type.

- (1) Register two Items that will store VT_I2 values (In this sample, “VT_2_Hi” and “VT_2_Lo”).
- (2) Set a link from VT_I4, which is the decomposition source, to the newly created items.
- (3) For “VT_I2_Hi” item, apply a binary operation to perform a 16 bit shift operation. For “VT_I2_Lo” item, apply binary operation to take logical AND with “65535 (&HFFFF)”.
- (4) After the above mentioned setting, execute CaoSQL. “15” will be stored in “VT_I2_Hi” and “65535” will be stored in “VT_I2_Lo”.



Data Conversion

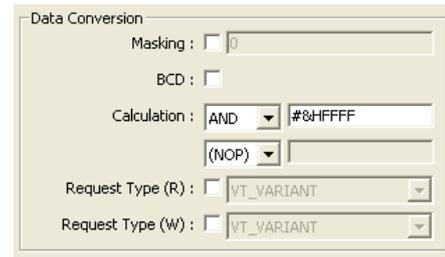
Masking : 0

BCD :

Calculation : #16

Request Type (R) : VT_VARIANT

Request Type (W) : VT_VARIANT

VT_I2_Hi

Data Conversion

Masking : 0

BCD :

Calculation : #&HFFFF

Request Type (R) : VT_VARIANT

Request Type (W) : VT_VARIANT

VT_I2_Lo**Figure 4-37 Binary operation setting for the decomposed items (detailed settings)**

4.6. Provider setting Tips

This section explains useful information for connecting various providers by using CaoSQLConfig.

4.6.1. DataStore provider

DataStore provider has an internal variable table, and ORiN2 applications can share data by sharing the variable table. For details, refer to DataStore provider user's guide.

Using DataStore provider in CaoSQL enables to collect data from other devices, to process or calculate the collected data, and then to store or transmit the processed or calculated result.

The following shows how to set Controller and Item.

(1) Controller setting

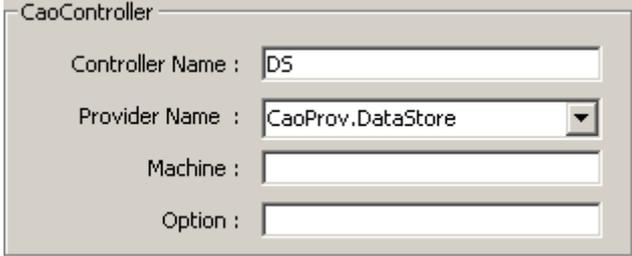


Figure 4-38 Controller setting of DataStore provider

- [Controller Name]
Enter arbitrary name. (Example: “DS”)
- [Provider Name]
Select “CaoProv.DataStore”.
- [Machine]
Specify the machine name to execute DataStore provider.
- [Option]
This is not used in DataStore provider.

(2) Item setting

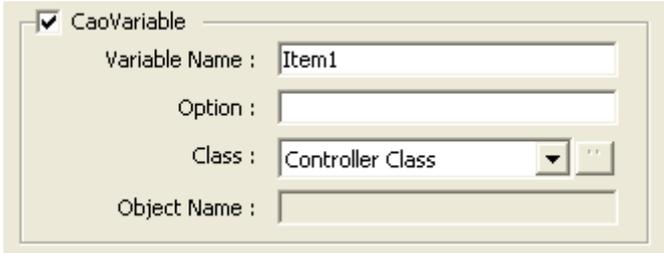


Figure 4-39 Item setting of DataStore provider

- [Variable Name]
Enter arbitrary name. (Example: “Item1”)
- [Option]
Specify option for AddVariable if necessary.
- [Class]
Specify “Controller Class”. This is because AddVariable of DataStore provider belongs to Controller class.
- [Object Name]
This is not used in DataStore provider.

4.6.1.1. Advanced usage of DataStore provider with Vars function

DataStore provider has Vars function. By acquiring data from a device and linking the data to Vars, the data can be passed to another application that is connected by Vars function. Another application of Vars is to apply calculations like binary operation to the values acquired from / written to a device, without altering the original value of the device. (Refer to Figure 4-40)

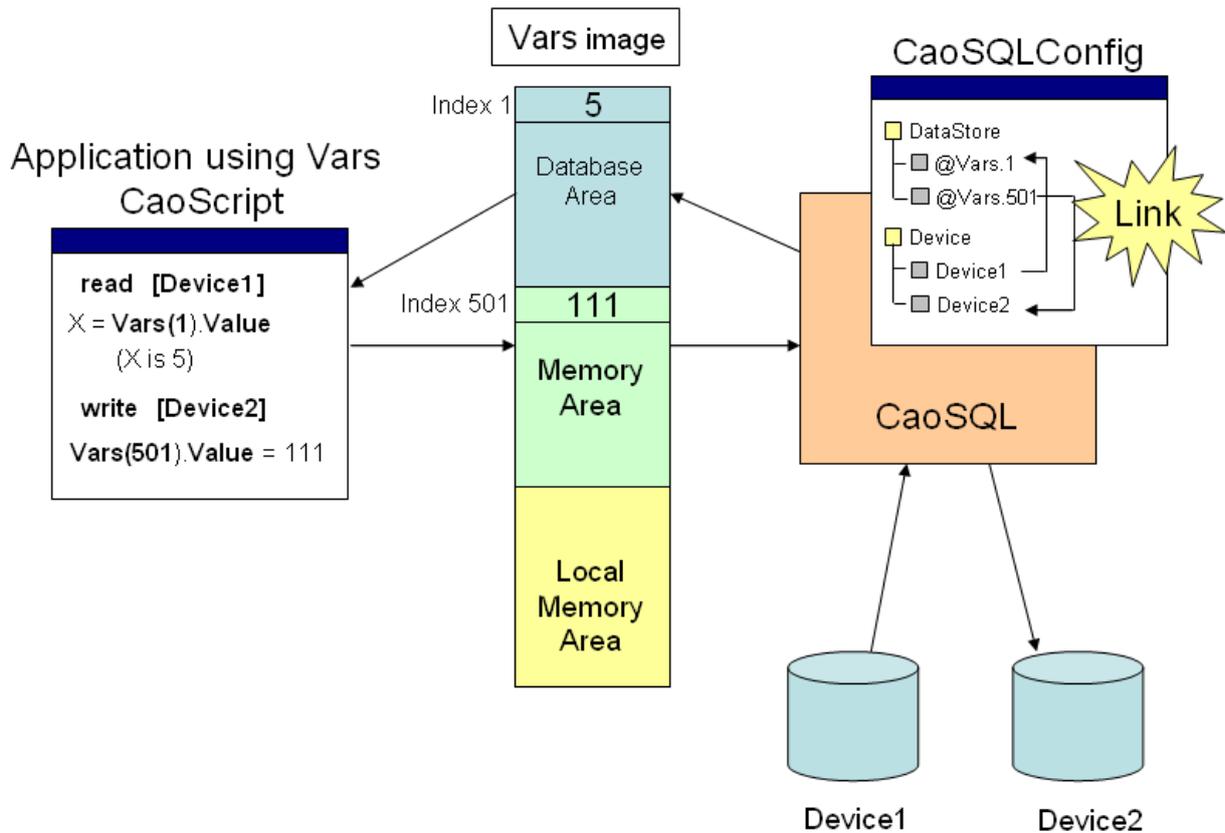


Figure 4-40 Example of using Vars function of DataStore provider

To use Vars in CaoSQLConfig, specify the Vars as ItemName: "@Vars.1" for the controller that uses DataStore provider. This allows to access the first element of Vars. For details, please refer to [DataStore provider user's guide](#).

4.6.2. DataBase provider

Using DataBase provider in CaoSQL enables to access any record in any field in the database. This function is practical to create a table in a database to store current value.

In the following example, DataBase provider uses "Sample.mdb" (Microsoft Access 2003)" database. "Sample.mdb" consists of a table [TestTable], two fields [FieldData1](text type, primary key) and [FieldData2](text type), and a field data [Test] where "FieldData1" has been set beforehand. Connection strings used are based on the setting for "Microsoft Access". For details, please refer to DataBase provider user's guide.

The following examples shows how to set up a controller and an item.

- (1) Controller settings

Figure 4-41 DataBase provider controller setting

- [Controller Name]: Specify the table name of the access destination database. (Example: TestTable)
- [Provider Name]: Choose “CaoProv.DataBase”.
- [Machine] : Specify a computer to run DataBase provider
- [Option] : Specify ADO connection string that connects to database..⁴⁰

(Example)

```
Provider=Microsoft.Jet.OLEDB.4.0,
Data Source=D:\ORiN2\CaoSQL\Bin\Sample.mdb,Key=FieldData1)
```

An error occurs if inappropriate table name or connection string is entered. Please pay attention to commas and spaces.

(2) Item settings

Figure 4-42 DataBase provider item setting

- [Variable Name]: Enter a field name. (Example: “FieldData2”)
- [Option]: Specify AddVariable options if necessary. (Example: “FieldData2”)
- [Class]: Specify “Extension Class” and then execute AddExtention.
- [Object Name]

To identify the target record, enter the value of Key field that has been specified in the connection string of the Controller setting. (Example: “Test”)

Specifying field name to “Key” of Controller connection string, and specifying key value in [Object

⁴⁰ For connection string to another database, please refer to [“DataBase provider user’s guide”](#).

Name] of item setting will enable to identify which record in the table is accessed by the DataBase provider.

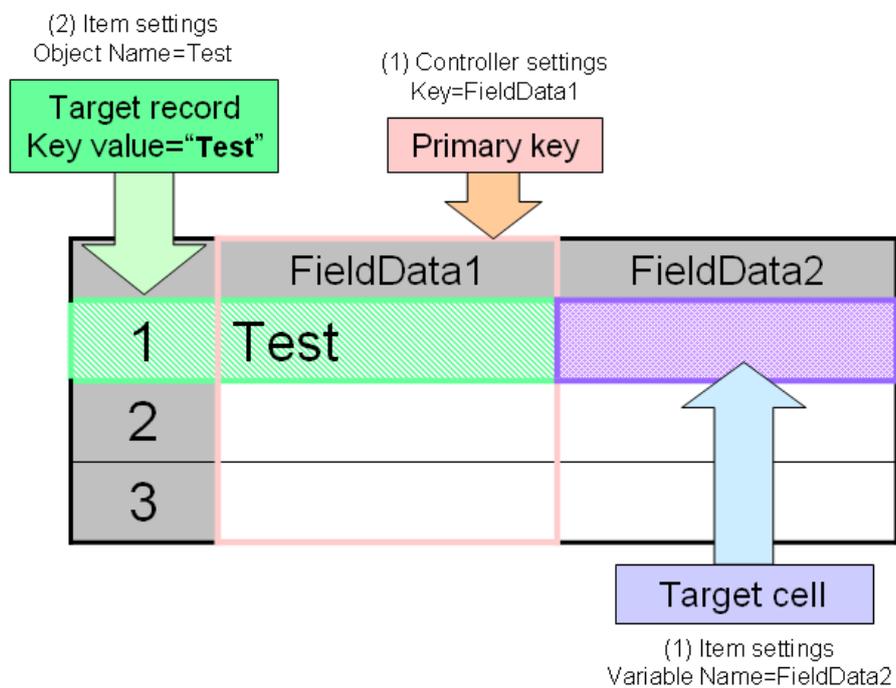


Figure 4-43 Target cell setting

When using DataBase provider in CaoSQL, an error occurs if the accessed data does not exist in the database. Although DataBase provider has a function of adding record, CaoSQL currently does not support this function.

5. CaoSQL tutorial

5.1. Outline

This chapter explains how to construct a system with CaoSQL. In this example, Access2000 and PostgreSQL are used.

5.2. Database setting by Access2000

ORiN2 SDK prepares the skeleton of Access2000 database file. If Jet Database engine is installed, the History function of CaoSQL is available even if Access2000 is not installed.

To use this Database file, start CaoSQLConfig, click "History" button, and set the items as follows.

- OLEDB Provider : Microsoft.Jet.OLEDB.4.0
- Data Source :
Absolute path of skeleton2000_en.mdb (stored in \CaoSQL\Bin of the ORiN2 SDK installation directory.)
- Table Name : caosql_history

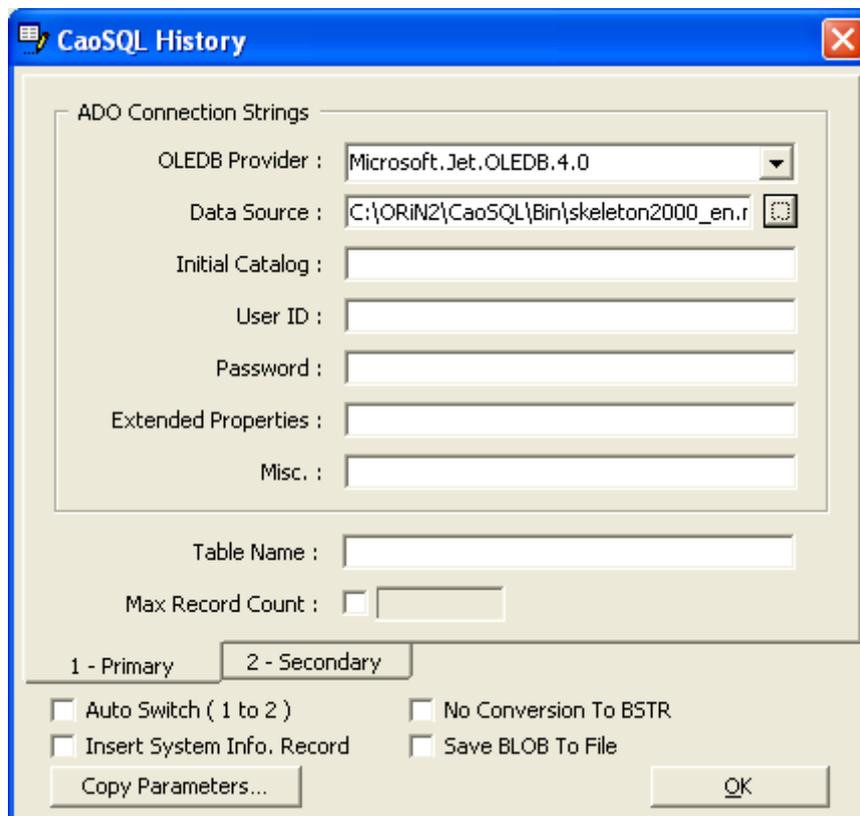


Figure 5-1 Access2000 database setting

5.3. Controller and Item registration by CaoSQLConfig

This section explains how to set a controller and item used in CaoSQL by using CaoSQLConfig. For further information for use of CaoSQLConfig, please refer to "[CaoSQLTools user's guide](#)".

Click "History" button of the "Engine" tab.

Next, set the controller. In this example, use DataStore provider to confirm the operation. From the menu bar, select [Edit] -[Add Controller]. Set the value of each item as Figure 5-2 shows.

Next, set the items. From the menu bar, select [Edit]-[Add Item], and then enter the item name. In this sample, "ItemA", "ItemB", and "ItemC" are entered.

This chapter explains the way of History function confirmation only. To use functions other than History function, please refer to Chapter 3 "Function of CaoSQL".

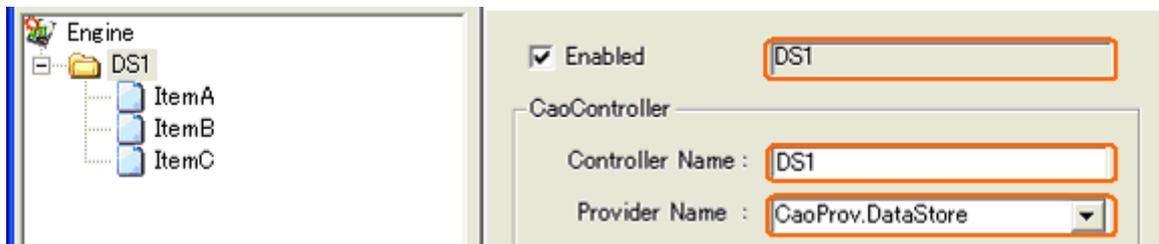


Figure 5-2 Adding a controller

5.4. Creating a client application

This section explains how to create a CaoSQL client application by using Visual Basic 6.0. From "New Project", select "Standard EXE".

(1) Add a Reference

Before using CaoSQL, you need to add a reference. From the menu bar, click [Project] - [Reference setting]. Add the following library file as Figure 5-3 shows.

- CaoSQL 1.0 type library
- Microsoft ActiveX Data Objects 2.7 Library

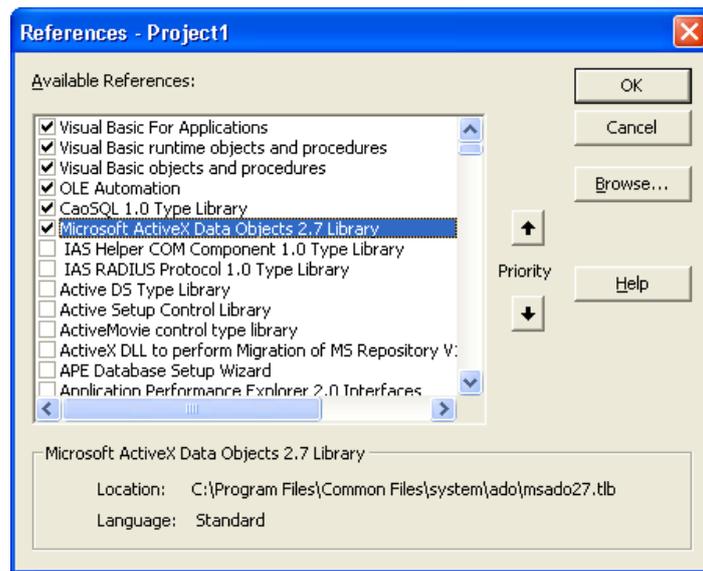


Figure 5-3 Reference setting

(2) Create a Form

Create a new form that has two combo boxes, three text boxes, and two command buttons as shown below. Red letters in the following figure represents the object name of each object.

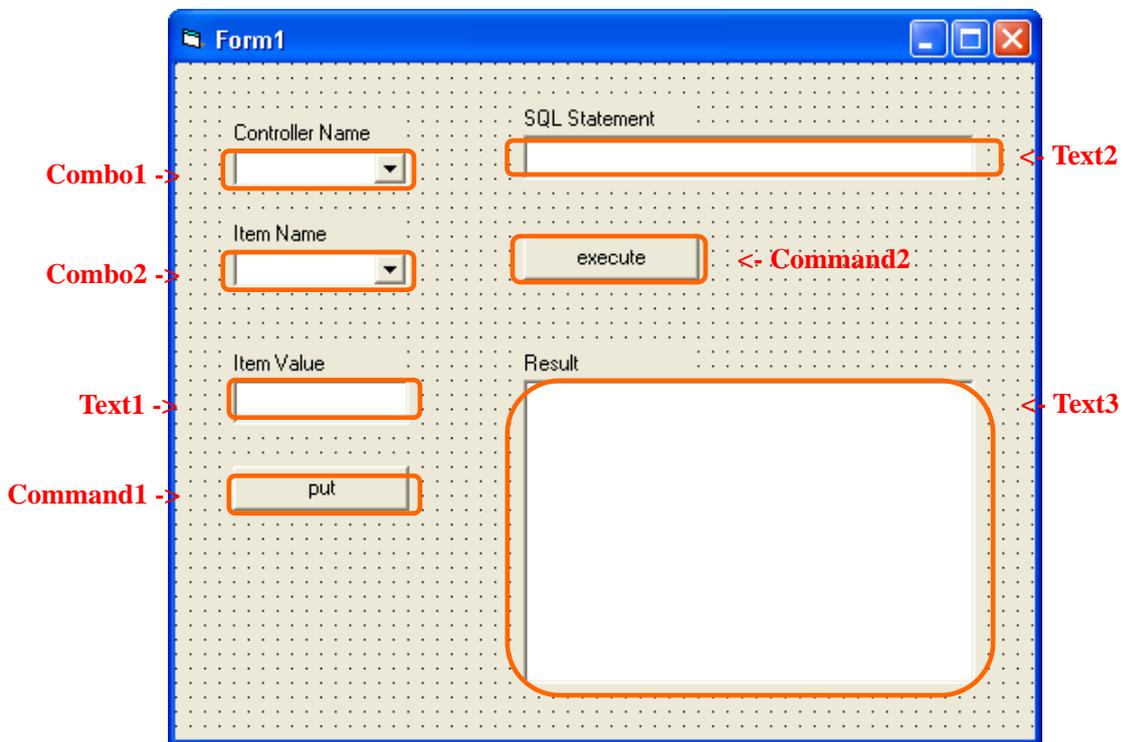


Figure 5-4 Client application form

(3) Write the source code

Write the following source code.

List 5-1	Form1.frm
-----------------	------------------

```

Private csqLEng As CaoSQLEngine      ' Engine of CaoSQL
Private csqLHis As CaoSQLHistory    ' Database of CaoSQL
Private csqLCtrl As CaoSQLController ' Controller of CaoSQL
Private csqLItem As CaoSQLItem      ' Item of CaoSQL
Private recSet As ADODB.Recordset    ' SQL sentence execution result

Private Sub Form_Load()
    Set csqLEng = New CaoSQLEngine
    Set csqLHis = csqLEng.CaoSQLHistory

    ' Acquire the registered controller-name and expand it to the combo box.
    Dim controllerName As Variant
    For Each controllerName In csqLEng.ControllerNames
        Combo1.AddItem (controllerName)
    Next
End Sub

Private Sub Combo1_Click()
    ' Acquire the selected controller
    Set csqLCtrl = csqLEng.Controller(Combo1.Text)
    ' Expand the item list of the selected controller to the combo box.
    Dim itemName As Variant
    For Each itemName In csqLCtrl.ItemNames
        Combo2.AddItem (itemName)
    Next
End Sub

Private Sub Combo2_Click()
    ' Acquire the selected item.
    Set csqLItem = csqLCtrl.Item(Combo2.Text)
    Command1.Enabled = True
End Sub

Private Sub Command1_Click()
    ' Write the value of textbox 1 into the item.
    csqLItem.Value = Text1.Text
End Sub

Private Sub Command2_Click()
    ' Execute SQL sentence of text box 2.
    Set recSet = csqLHis.Execute(Text2.Text)
    ' Display the execution result into the text box 3.
    Dim cnt As Integer
    Text3.Text = ""
    Do Until recSet.EOF
        For cnt = 0 To recSet.Fields.Count - 1
            Text3.Text = Text3.Text & " " & recSet(cnt).Value
        Next
        Text3.Text = Text3.Text & vbNewLine
        recSet.MoveNext
    Loop
End Sub

```

5.5. Execution of sample

(1) Start of CaoSQL

Start CaoSQLLauncher.exe, and click "Start" button.

(2) Start of client application

Start the client application created in 5.4.

If the application starts successfully, the following window appears. Click "Controller name" combo box to display the list of controller names that have been set with CaoSQLConfig.

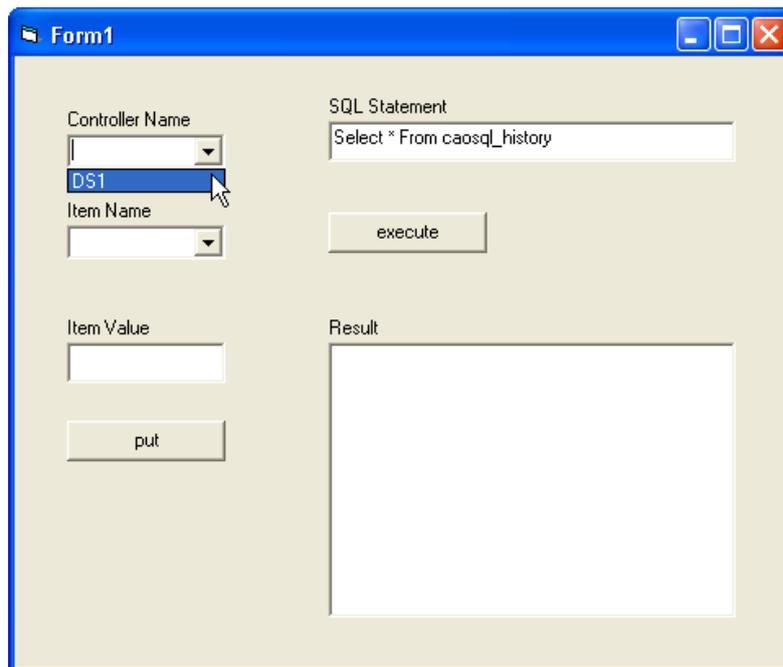
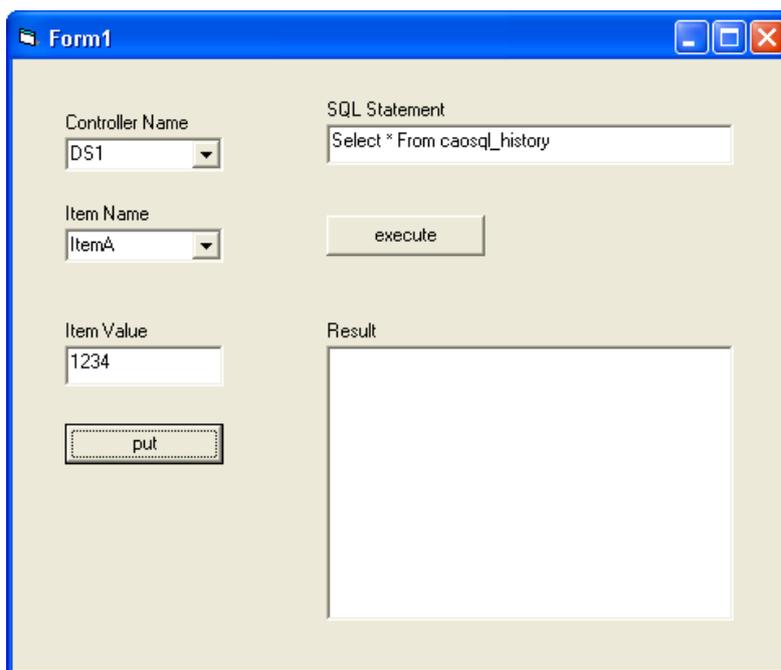


Figure 5-5 Start of client application

(3) Writing data

Select appropriate controller-name and item name from the each combo box as shown in Figure 5-6, enter the value to the item value text box, and then click "put" button.

This step simply changes the item value of controller though, the changed item is registered in the database by History function.



Form1

Controller Name: DS1

SQL Statement: Select * From caosql_history

Item Name: ItemA

execute

Item Value: 1234

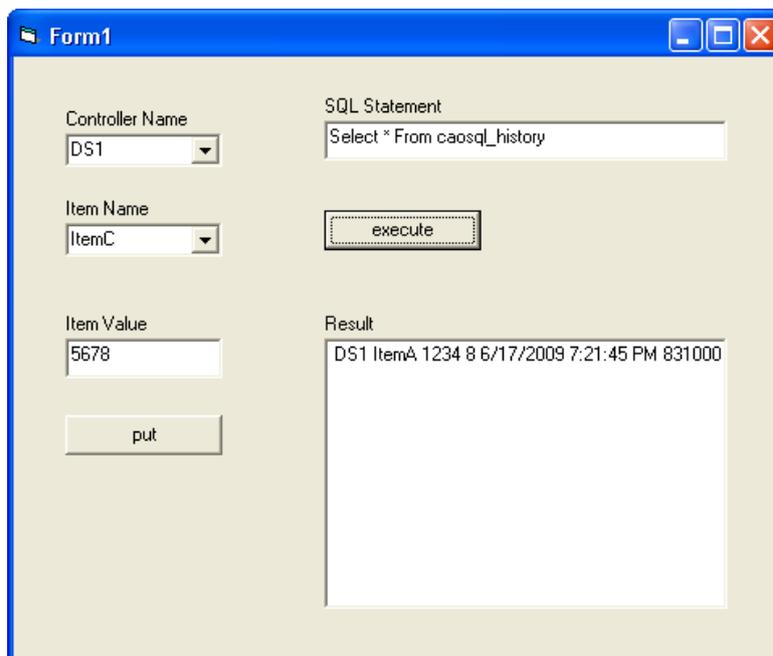
put

Result

Figure 5-6 Writing item

(4) Executing SQL sentence

Click [Execute] button to execute SQL sentence of “ Select * From caosql_history”. The data list that have been written just before is displayed in the text box.



Form1

Controller Name: DS1

SQL Statement: Select * From caosql_history

Item Name: ItemC

execute

Item Value: 5678

put

Result

```
DS1 ItemA 1234 8 6/17/2009 7:21:45 PM 831000
```

Figure 5-7 Execution of SQL sentence

Appendix A.

Appendix A.1. CaoSQL API list

- CaoSQLEngine Object-engine

Object	Property, Method and Event		Description	R/W /S	Argument of function		Remarks
					IN	OUT RETVAL	
CaoSQLEngine	Controller	P	Acquisition of controller object	R	Controller name/controller number: VARIANT	Controller object: ICaoSQLController	Default member
	Count	P	Acquisition of number of controllers	R		Number of controllers: Long	
	ControllerNames	P	Acquisition of controller name list	R		Controller name list: VARIANT (VT_ARRAY VT_BSTR)	
	CaoSQLHistory	P	Acquisition of history object	R		History object: ICaoSQLHistory	
	Execute	M	Execution of extended command	S	Command: VARIANT	Execution result: VARIANT	For function extension
	Snapshot	M	Save current values of all items in all controllers into history DB.	W			
	Start	M	Start all sampling threads	W			
Stop	M	Stop all sampling threads	W				

Value of a symbol	M:Method P:Property E:Event		(note 1)	<ul style="list-style-type: none"> ·Arguments enclosed by square brackets “[]” can be omitted. ·The default value of BSTR type argument that can be omitted is empty string. ·The default value of numeric type argument that can be omitted is 0. 	
-------------------	-----------------------------------	--	----------	--	--

(note 1)The value of a symbol is as follows.

R-Read: Acquire status and the configuration of the controller or the provider.

W-Write: Change controller's status and configuration.

S-Setup: Change status and the configuration of the provider.

- CaoSQLController Object-controller

Object	Property, Method and Event		Description	R/W /S	Argument of function		Remarks
					IN	OUT RETVAL	
CaoSQLController	Count	P	Acquisition of number of items	R		Number of items: Long	
	Description	P	Details of a controller	R		Details: BSTR	
	Index	P	Acquisition of item number	R		Item number: Long	
	Item	P	Acquisition of item object	R	Item name/item number: VARIANT	Object: ICaoSQLItem	Default member
	ItemNames	P	Acquisition of item name list	R		Item name list: VARIANT (VT_ARRAY VT_BSTR)	
	Name	P	Controller name	R		Controller name: BSTR	
	ScanTimes	P	Acquisition of the scan time	R	Scanning time type number: Long	Scan time: Long	For about the scan time type number, refer to "enmCaoSQLCtrlScantime".
	SettingData	P	Acquisition of setting data	R	Data number/data name: VARIANT	Data: VARIANT	For Data name/Data number., refer to "get_SettingData argument list". PutElement has been reserved.

State	P	The controller's state acquisition	R		Status: Long	Refer to enmCaoSQLCtrlState.
Tag	P	Tag	R/W	Tag: BSTR	Tag: BSTR	Multipurpose tag
AddItem	M	Dynamic addition of item	W	Item data: VARIANT		
RemoveItem	M	Deletion of item	W	Item name/Item number: VARIANT		Only the item dynamically added by AddItem can be deleted.
Snapshot	M	Save the current value of all items into History DB.	W			
Start	M	Start a sampling thread	W			
Stop	M	Stop a sampling thread	W			
Execute	M	Execution of command character string	S	Command: BSTR Parameter: VARIANT		Pass the command character string to Execute of CaoController, and execute it.
OnChangeItem	E	Value change event	-		CaoSQLItem interface pointer: ICaoSQLItem *	
OnChangeState	E	Status change event	-		Status value: Long	Refer to enmCaoSQLCtrlState for the status value.

Value of a symbol	M:Method P:Property E:Event		(note 1)	<ul style="list-style-type: none"> ·Arguments enclosed by square brackets “[]” can be omitted. ·The default value of BSTR type argument that can be omitted is empty string. ·The default value of numeric type argument that can be omitted is 0. 	
-------------------	-----------------------------------	--	-------------	--	--

- enmCaoSQLCtrlState

Enum	Remarks
0: csCtrlThreadUninit	The thread is not initialized.
1: csCtrlThreadActive	The thread is running.
2: csCtrlThreadDeactive	The thread is stop-state.
3: csCtrlThreadError	An error is occurred by the thread.
4: csCtrlThreadTerminated	The thread terminated.

- enmCaoSQLCtrlScantime

Enum	Remarks
0: csCtrlScantimeLast	The latest scanning time
1: csCtrlScantimeMin	Minimum scanning time
2: csCtrlScantimeMax	Maximum scanning time

- Get_SettingData argument list

Property number	Property name	Description
0	Controller Name	Controller name

1	Provider Name	Provider name
2	Machine	Computer name
3	Option	Option
4	Use History	Enable/Disable the automatic recording of History
5	Write to Cache	Enable/Disable of cache writing function
6	Time Stamp	Timestamp
7	Sampling Rate	Sampling interval
8	Distribute Equally	Enable/Disable the even distribution
9	Startup	Enable/Disable the Start-up registration
10	Priority	Thread priority
11	(n/a)	Enable/Disable the Watch dog timer
12	(n/a)	Watch dog timer counter
13	(n/a)	Watch dog timer action
14	(n/a)	Enable/Disable the Automatic reconnection
15	(n/a)	Automatic reconnection condition
16	(n/a)	Automatic reconnection-retry interval
17	(n/a)	Type of controller
18	(n/a)	Enable/Disable the script
19	(n/a)	Script source
20	(n/a)	(unpublished)

- CaoSQLItem Object-item

Object	Property, Method and Event	Description	R/W /S	Argument of function		Remarks	
				IN	OUT RETVAL		
CaoSQLItem	Activation	P	Acquisition/Setting of Item value change event ON/OFF state	R/W	Setting: VARIANT_BOOL	Setting: VARIANT_BOOL	
	Attribute	P	Acquisition of item attribute	R		Item attribute: Long	Refer to enmCaoSQLItemAttribute.
	DateTime	P	Acquisition of time and date stamp (date)	R		Time and date stamp: VARIANT	
	Description	P	Details of a controller	R		Details: BSTR	
	Element	P	Value acquisition of array element	R	1st dimension element: long 2nd dimension element: long 3rd dimension element: long	Value: VARIANT	PutElement has been reserved.
	ID	P	Item ID	R/W	Item ID:VARIANT	Item ID: VARIANT	
	Index	P	Item number	R		Item number: Long	
	LinkActivation	P	Acquisition/Setting of Link destination's value change event ON/OFF state	R/W	Setting: VARIANT_BOOL	Setting: VARIANT_BOOL	

LinkItemNames	P	Link destination item name	R		Item name of link destination: VARIANT(VT_ARRAY VT_BSTR)	The format is as follows. “[<LinkControllerName>:]<LinkItemName>” Retrieve an Item from the current controller when “LinkControllerName” is omitted.
Microsecond	P	Acquisition of time and date stamp (microsecond)	R		Time and date stamp: Long	
Name	P	Acquisition of item name	R		Item name: BSTR	
SettingData	P	Acquisition of each setting data	R	Data number/Data name: VARIANT	Data: VARIANT	For Data number/Data name, refer to the get_SettingData argument list. PutSettingData has been reserved.
State	P	Acquisition of item's state	R		Status: Long	Refer to enmCaoSQLItemState.
Tag	P	Tag	R/W	Tag: VARIANT	Tag: VARIANT	Multipurpose tag
Value	P	Acquisition/Setting of item's value	R/W	Value: VARIANT	Value: VARIANT	Default member
Snapshot	M	Save the current value into History DB.	W			

Value of a symbol	M:Method P:Property E:Event		(note 1)	<ul style="list-style-type: none"> · Arguments enclosed by square brackets “[]” can be omitted. · The default value of BSTR type argument that can be omitted is empty string. · The default value of numeric type argument that can be omitted is 0. 	
-------------------	-----------------------------------	--	----------	---	--

- enmCaoSQLItemState

Enum	Remarks
0: csItemSucceeded	Value setting/acquisition of the item succeeded.
1: csItemFailed	Value setting/acquisition of the item failed.
2: csItemUndefined	The value of the item is undefined.

- enmCaoVariableClass

Enum	Remarks
0: csVarClassController	Variable of controller class
1: csVarClassTask	Variable of task class
2: csVarClassRobot	Variable of robot class
3: csVarClassExtension	Variable of extension class

- enmCaoSQLItemAttribute

Enum	Remarks
0: csItemReadWrite	The item is reading/writing attribute.

1: csItemReadOnly	The item is read-only.
2: csItemWriteOnly	The item is write-only.

- get_SettingData argument list

Property number ⁴¹	Property name	Description
0	Variable Name	CaoVariable name
1	Option	Option for creating CaoVariable
2	Class	Parent class type of CaoVariable 0 Controller class 1 Task class 2 Robot class 3 Extension class 4 File class
3	Object Name	Parent object name of CaoVariable
4	Initialized	Enable/Disable Initialization
5	Initial Data	Initialization value

⁴¹ Note that the property number is not same as the array element number described in [Table 3-11 Argument to AddItem method]. For the array element number of 3 or less, [Property number] + 2 will be the array element number. For the array element number of 4 or larger, [the property number] + 5 will be the array element number.

6	Initial Data Type	Initialization type (VARTYPE) 2 VT_I2 3 VT_I4 4 VT_R4 5 VT_R8 6 VT_CY 7 VT_DATE 8 VT_BSTR 11 VT_BOOL 12 VT_VARIANT 16 VT_I1 17 VT_UI1 18 VT_UI2
7	Use Empty	Set whether to use VT_EMPTY as an effective data type or not
8	History	Set whether to use History function
9	OnChange Event	Enable/Disable CaoSQL item event function
10	Force to Change	Forcible value change setting. 0 None 1 At the time of reading 2 At the time of writing
11	(n/a)	Enable/Disable the Dead band setting
12	(n/a)	Minimum value of dead band
13	(n/a)	Maximum value of dead band
14	(n/a)	Enable/Disable the valid range setting
15	(n/a)	Minimum value of valid range
16	(n/a)	Maximum value of valid range
17	(n/a)	Enable/Disable the Chattering setting
18	(n/a)	Chattering value

19	(n/a)	Enable/Disable the Masking setting																																																								
20	(n/a)	Mask value																																																								
21	(n/a)	Enable/Disable BCD conversion setting																																																								
22	(n/a)	<p>Binary operation processing operator (the first operation)</p> <table> <tr> <td>0</td> <td>(NOP)</td> <td>1</td> <td>+</td> </tr> <tr> <td>2</td> <td>-</td> <td>3</td> <td>*</td> </tr> <tr> <td>4</td> <td>/</td> <td>5</td> <td>MOD</td> </tr> <tr> <td>6</td> <td>AND</td> <td>7</td> <td>OR</td> </tr> <tr> <td>8</td> <td>XOR</td> <td>9</td> <td><<</td> </tr> <tr> <td>10</td> <td>>></td> <td>11</td> <td>=</td> </tr> <tr> <td>12</td> <td><<</td> <td>13</td> <td><=</td> </tr> <tr> <td>14</td> <td><</td> <td>15</td> <td>>=</td> </tr> <tr> <td>16</td> <td>></td> <td>17</td> <td>CAT</td> </tr> <tr> <td>18</td> <td>IDIV</td> <td>19</td> <td>IMP</td> </tr> <tr> <td>20</td> <td>POW</td> <td>21</td> <td>ABS</td> </tr> <tr> <td>22</td> <td>FIX</td> <td>23</td> <td>INT</td> </tr> <tr> <td>24</td> <td>NEG</td> <td>25</td> <td>NOT</td> </tr> <tr> <td>26</td> <td>RAD2DEG</td> <td>27</td> <td>DEG2RAD</td> </tr> </table>	0	(NOP)	1	+	2	-	3	*	4	/	5	MOD	6	AND	7	OR	8	XOR	9	<<	10	>>	11	=	12	<<	13	<=	14	<	15	>=	16	>	17	CAT	18	IDIV	19	IMP	20	POW	21	ABS	22	FIX	23	INT	24	NEG	25	NOT	26	RAD2DEG	27	DEG2RAD
0	(NOP)	1	+																																																							
2	-	3	*																																																							
4	/	5	MOD																																																							
6	AND	7	OR																																																							
8	XOR	9	<<																																																							
10	>>	11	=																																																							
12	<<	13	<=																																																							
14	<	15	>=																																																							
16	>	17	CAT																																																							
18	IDIV	19	IMP																																																							
20	POW	21	ABS																																																							
22	FIX	23	INT																																																							
24	NEG	25	NOT																																																							
26	RAD2DEG	27	DEG2RAD																																																							
23	(n/a)	Binary operation processing operation value setting (the first operation)																																																								
24	(n/a)	Enable/Disable the data type conversion processing. (for reading)																																																								

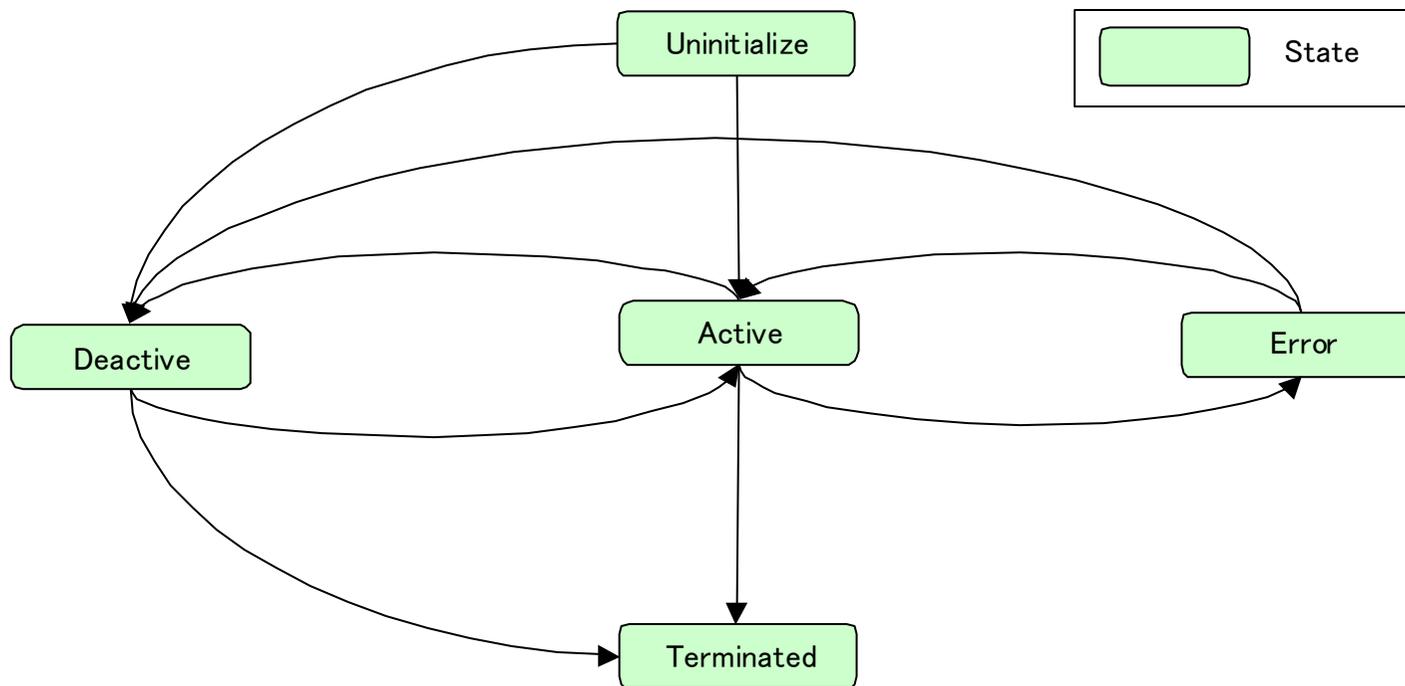
25	(n/a)	Data type setting of data type conversion processing. (Read, VARTYPE) <table> <tr> <td>2</td> <td>VT_I2</td> <td>3</td> <td>VT_I4</td> </tr> <tr> <td>4</td> <td>VT_R4</td> <td>5</td> <td>VT_R8</td> </tr> <tr> <td>6</td> <td>VT_CY</td> <td>7</td> <td>VT_DATE</td> </tr> <tr> <td>8</td> <td>VT_BSTR</td> <td>11</td> <td>VT_BOOL</td> </tr> <tr> <td>12</td> <td>VT_VARIANT</td> <td>16</td> <td>VT_I1</td> </tr> <tr> <td>17</td> <td>VT_UI1</td> <td>18</td> <td>VT_UI2</td> </tr> <tr> <td>19</td> <td>VT_UI4</td> <td></td> <td></td> </tr> </table>	2	VT_I2	3	VT_I4	4	VT_R4	5	VT_R8	6	VT_CY	7	VT_DATE	8	VT_BSTR	11	VT_BOOL	12	VT_VARIANT	16	VT_I1	17	VT_UI1	18	VT_UI2	19	VT_UI4		
2	VT_I2	3	VT_I4																											
4	VT_R4	5	VT_R8																											
6	VT_CY	7	VT_DATE																											
8	VT_BSTR	11	VT_BOOL																											
12	VT_VARIANT	16	VT_I1																											
17	VT_UI1	18	VT_UI2																											
19	VT_UI4																													
26	(n/a)	Enable/Disable the data type conversion processing. (for writing)																												
27	(n/a)	Data type setting of data type conversion processing. (for writing, VARTYPE) <table> <tr> <td>2</td> <td>VT_I2</td> <td>3</td> <td>VT_I4</td> </tr> <tr> <td>4</td> <td>VT_R4</td> <td>5</td> <td>VT_R8</td> </tr> <tr> <td>6</td> <td>VT_CY</td> <td>7</td> <td>VT_DATE</td> </tr> <tr> <td>8</td> <td>VT_BSTR</td> <td>11</td> <td>VT_BOOL</td> </tr> <tr> <td>12</td> <td>VT_VARIANT</td> <td>16</td> <td>VT_I1</td> </tr> <tr> <td>17</td> <td>VT_UI1</td> <td>18</td> <td>VT_UI2</td> </tr> <tr> <td>19</td> <td>VT_UI4</td> <td></td> <td></td> </tr> </table>	2	VT_I2	3	VT_I4	4	VT_R4	5	VT_R8	6	VT_CY	7	VT_DATE	8	VT_BSTR	11	VT_BOOL	12	VT_VARIANT	16	VT_I1	17	VT_UI1	18	VT_UI2	19	VT_UI4		
2	VT_I2	3	VT_I4																											
4	VT_R4	5	VT_R8																											
6	VT_CY	7	VT_DATE																											
8	VT_BSTR	11	VT_BOOL																											
12	VT_VARIANT	16	VT_I1																											
17	VT_UI1	18	VT_UI2																											
19	VT_UI4																													
28	(n/a)	Enable/Disable the data comparison in array																												
29	(n/a)	Array element distribution link setting																												
30	(n/a)	Inactivation																												

31	(n/a)	Item type 0 General item 1 Alias item 2 Array type item
32	(n/a)	Link destination's property 0 Value 1 ID 2 Activation 3 LinkActivation
33	(n/a)	Binary operation processing operator (the first operation)
34	(n/a)	Binary operation processing value setting (the second operation)

- CaoSQLHistoryObject-history

Object	Property, Method and Event		Description	R/W /S	Argument of function		Remarks
					IN	OUT RETVAL	
CaoSQLHistory	ConnectionString	P	ADO connection character string of History	R	Primary /Secondary setting: long	Connected character string: BSTR	Enter "1" for primary, "2" for secondary.
	Enable	P	Enable/Disable the History function	R		Enable/Disable History setting: VARIANT_BOOL	
	MaxRecord	P	Number of record limit	R	Primary /Secondary setting: long	Number of maximum records: Long	Enter "1" for primary, "2" for secondary. If "0" is returned, the number of record limit is not specified..
	TableName	P	Table name	R	Primary /Secondary setting: long	Table name: BSTR	Enter "1" for primary, "2" for secondary.
	Execute	M	Execution of SQL sentence	-	SQL sentence: BSTR	ADO::Recordset: VARIANT	Execute the entered SQL sentence as it is.
Value of a symbol	M:Method P:Property E:Event			(note 1)	<ul style="list-style-type: none"> ·Arguments enclosed by square brackets "[]" can be omitted. ·The default value of BSTR type argument that can be omitted is empty string. ·The default value of numeric type argument that can be omitted is 0. 		

Appendix A.2. CaoSQLController state transition chart



Appendix A.3. CaoSQLController state transition chart

