Database provider

Database access

Version 1.0.2

User’s guide

September 15, 2023

[Remarks]

[Revision history]

Version	Date	Contents
1.0.0.0	2006-02-23	First edition
1.0.0.1	2010-02-10	Added error codes.
1.0.0	2012-07-17	Modified the version rule of document management.
	2015-08-03	Added RecordsetToArray to AddController option.
	2015-08-05	Added SQL to AddController option
	2015-08-26	Added BeginTrans, CommitTrans, and RollbackTrans to Execute commands.
	2015-09-02	Added Requery to Execute commands.
1.0.1	2016-02-19	Added Misc to option character strings of AddController.
	2017-06-29	Added connection example of Access 2017 (*. Accdb).
1.0.2	2018-10-30	Memory leak was corrected.
	2018-12-05	Added Appendix A.
	2019-12-26	Modified document, Acess -> Access
	2020-12-01	Added PostgreSQL connection example. Added @Connected to AddController option. Fixed Oracle connection example. Added connection example of Excel2007 or later (* .xlsx, * .xlsm, * .xlsb). Added description of PostgreSQL.
	2022-11-21	Added description of MySQL.
	2023-04-11	Error Correction.
	2023-09-15	Error Correction.

[Supported devices]

Device	Version	Notes

Contents

1. Introduction 4
2. Outline of provider 5
2.1. Outline 5
2.1.1. Add a record 6
2.1.2. Delete a record 6
2.2. Method and Properties 7
2.2.1. CaoWorkspace::AddController method 7
2.2.2. CaoController::AddExtension method 11
2.2.3. CaoController::Execute method 12
2.2.4. CaoExtension::AddVariable method 12
2.2.5. CaoExtension::get_VariableNames property 12
2.2.6. CaoExtension::Execute method 13
2.2.7. CaoVariable::get_Value method 13
2.2.8. CaoVariable::put_Value method 13
2.3. Variable list 13
2.4. Error code 13
3. Sample program 14
4. Command reference 15
4.1. Command list 15
4.2. Command descriptions 15
Appendix A. About file name and table name 18

1. Introduction

This is a user’s guide of Database provider (hereafter, DB provider) that accesses database by means of CAO interface.

DB provider is designed to facilitate data control by using CAO interface. Database is controlled via ADO interface and is mounted by wrapping ADO in CAO interface for connection. In general, DB provider also supports OLEDB provider-supported database engine.

This document describes the functions of DB provider and the implemented methods.

2. Outline of provider

2.1. Outline

Table 2-1 shows the correspondence between Database and CAO class. CaoController, CaoExtension, and CaoVariable are assigned to Database's Table, Record, and Cell, respectively. This correspondence enables to specify any cell position of database by specifying a table name, key value, and cell name of each object.

Table 2-1 Correspondence between CAO class and Database

Class	Database
Controller	Table
Extension	Record
Variable	Cell

DB provider implements Execute method of CaoController object. This object is connected to Execute method of ADO.Connection object so that SQL sentence can be executed via CAO interface.

The file format of DB provider is DLL (Dynamic Link Library) and Table 2-2 shows the details.

Table 2-2 DataBase provider

File name	CaoProvDB.dll
ProgID	CaoProv.DataBase
Registration[1]	regsvr32 CaoProvDB.dll
Deregistration	regsvr32 /u CaoProvDB.dll

For DB provider, MDAC (Microsoft Data Access Components)2.7 is required.

[1] You do not have to register/deregister it manually if the provider is installed by ORiN2 SDK.

2.1.1. Add a record

To add a record, in @Create option of CaoController::AddExtension method, enter 1 or larger value. At that time, an extension board name will be the key value of an added record. Because cells other than key cells cannot contain data, if any cell that needs value entry exists, it will fail to add a record.

2.1.2. Delete a record

To add a record, in a command of CaoExtension::Execute method, enter "Delete". Note that an object that connects to a deleted record and/or deleted cell are not deleted automatically after a record has been deleted. Please delete such objects by using client program.

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

At the timing of AddController (when creating an Controller object), DB provider creates an ADO.Connection object and connects to Database engine. At that time, ADO connection character strings is not modified by DB provider and is directly passed to Open method of ADO.Connection object.

The following shows an example when AddController method is executed.

```
AddController
(
    "<Controller name>",            // Table name.
    "CaoProv.DataBase ",            // Provider name. Fixed.
    "<Computer name>",              // Computer name where provider runs.
    "<Option>"            // Option character strings
)
```

The following table shows the list of option character strings.

Table 2-3 Option character strings of CaoWorkspace::AddController

Option	Description
Provider=<OLEDB provider name >	OLEDB provider of Database to open. (Table 2-4)
Data Source=<Data source >	Data source. (Table 2-4)
Initial Catalog=<Catalog name>	Initial catalog (Table 2-4)
User ID=<User ID>	User ID of Database. (Table 2-4)
Password=<Password >	Password of "User" (Table 2-4)
Extended Properties=<Attribute>	Additional Attribute. (Table 2-4)
Key[=<Key name>]	Field name that is used as a key, (Default: The primary key of the specified table.) Compound keys (keys spanning two or more columns) are not supported by DB providers.
RecordsetToArray[=<True/False>]	Execution result format of SQL command of CaoController::Execute method. True:VARIANT array False:Recordset object (Default)
SQL[=<SQL sentence>]	Replace a Recordset object used in provider with an SQL sentence execution result. (Default: A Recordset object that is obtained by using Controller name as a Table name will be used.)
Misc[=<Character strings to be added to ADO connection character strings	Character strings to be added to ADO connection character strings. Character strings must be enclosed with brackets as always.

>]	Example) Misc=(Database=DB;Port=2345)
@Connected[=<True/False>]	Data access method to database True: Connected data access (default) Each process is executed while maintaining the connection with the database. False: Disconnected data access Each process is executed without maintaining the database connection at all times.

Table 2-4 AddController setting list for different database

Database	Controller name	Option	Content to enter
SQL Server	Table name	Provider	"SQLOLEDB.1"
		Data Source	SQL server name
		Initial Catalog	Database name
		User ID	User ID of Database
		Password	Password of "User"
Oracle	Table name	Provider	"OraOLEDB.Oracle"[2]
		Data Source	Oracle server name
		User ID	User ID of Database
		Password	Password of "User"
Access97	Table name	Provider	"Microsoft.Jet.OLEDB.3.51"
		Data Source	File name (*.mdb)
Access2000	Table name	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	File name (*.mdb)
Access 2007 or later (*.ACCDB)	Table name	Provider	" Microsoft.ACE.OLEDB.12.0 "
		Data Source	File name (*.accdb)
Excel2000	[<sheet name>$<range>]	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	File name (*.xls)
		Extended Properties	"Excel 8.0"[3]
Excel2007 or later (*.xlsx, *.xlsm, *.xlsb)	[<sheet name>$<range>]	Provider	"Microsoft.ACE.OLEDB.12.0 "
		Data Source	File name (*.xlsx, *.xlsm, *.xlsb)
		Extended Properties	"Excel 12.0"[4]
CSV file	CSV file name	Provider	"Microsoft.Jet.OLEDB.4.0"
		Data Source	Folder path where CVS file exists.
		Extended Properties	"Text"
PostgreSQL[5]	Table name	Provider	"MSDASQL"

[2] It can be used to access the Oracle Database (Oracle 9.2 and above). For earlier Oracle Databases, use "MSDAORA".
[3] A blank shall be entered between "Excel" and "8.0".
[4] A blank shall be entered between "Excel" and "12.0".
[5] If uppercase letters are used in table and field names, you cannot connect to the target database server.

	Connection parameters	Misc	Server=<IP address>; Database=<DB name>; Uid=<User name>; Pwd=<Password>; Driver=PostgreSQL Unicode;
MySQL[5]	Table name	Provider	"ODBC"
	Connection parameters	Misc	Server=< IP address >; Database=< DB name >; Uid=< User name >; Pwd=< Password >; Driver=MySQL ODBC 5.3 Unicode Driver;

The following tables show examples of AddController for respective database.

Example 1 : For SQL server

Server name : TestServer

Database name : MyDB

Table name : TestTable

User name : User

Password : Pass

```
AddController("TestTable", "CaoProv.DataBase ", "", _
              "Provider=SQLOLEDB.1, Data Source=TestServer, Initial Catalog=MyDB, _
               User ID=User, Password=Pass")
```

Example 2 : For Oracle

Server name : TestServer

Database name : MyDB

Table name : TestTable

User name : User

Password : Pass

```
AddController("TestTable", "CaoProv.DataBase ", "", _
              "Provider=MSDATAORA, Data Source=TestServer, Initial Catalog=MyDB, _
               User ID=User, Password=Pass")
```

Example 3 : For Access2000

MDB file : C:¥TestDB.mdb

Table name : TestTable

```
AddController("TestTable", "CaoProv.DataBase ", "", _
              "Provider=Microsoft.Jet.OLEDB.4.0, Data Source= C:¥TestDB.mdb ")
```

Example 4 : For Excel2000

XLS file	:	C:¥TestDB.xls
Sheet name	:	Sheet1
Cell range	:	A1:B10

```
AddController("[Sheet1$A1:B10]", "CaoProv.DataBase ", "", _
              "Provider=Microsoft.Jet.OLEDB.4.0, Data Source=C:¥TestDB.xls, _
               Extended Properties=Excel 8.0")
```

Example 5 : For CSV file

CSV file	:	C:¥TestDB.csv

```
AddController("TestDB.csv", "CaoProv.DataBase ", "", _
              "Provider=Microsoft.Jet.OLEDB.4.0, Data Source=C:¥, _
               Extended Properties=Text")
```

Example 6 : For PostgreSQL

Server name	:	TestServer
Database name	:	MyDB
Table name	:	test_table

```
AddController("test_table", "CaoProv.DataBase", "", _
              "Provider=MSDASQL, Misc=(Server=TestServer;Database=MyDB;Uid=user; _
              Pwd=Pass; Driver=PostgreSQL Unicode;")
```

Example 7 : For MySQL

Server name	:	TestServer
Database name	:	MyDB
Table name	:	test_table

```
AddController("test_table", "CaoProv.DataBase", "", _
              "Provider=ODBC, Misc=(Server=TestServer;Database=MyDB;Uid=user; _
               Pwd=Pass; Driver=MySQL ODBC 5.3 Unicode Driver")
```

2.2.2. CaoController::AddExtension method

This method creates an extension board object that corresponds to a record in a table. The relation between a record and an extension board object is established by specifying a key value as an extension board name.

This method will fail if a table does not have a key that corresponds to an extension board name. Also, if two or more keys that correspond to an extension board name exist in a table, the smallest number's record will be

assigned to it..

To create a new record, specify @Create option. If the same extension board name has been exist in the table already, a new table is not created and the data you newly enter will be related to the existing record. This method will fail if any field requires a value entry into a table.

The following shows an example to execute AddExtension method.

```
AddExtension
(
    "<extension board name >",              // Key value
    "<Option>"           // Option character strings
)
```

The following shows a list to enter in option character strings.

Table 2 Option character strings of CaoController::AddExtension

Option	Description
@Create[=<0 to 1>]	・If there is no record specified by Extension board name, a record will be added based on this option value. 0: Do not add a record (Default) 1: Add a record

2.2.3. CaoController::Execute method

Execute a command of Controller class.

For details of each command, refer to Chapter 4.

2.2.4. CaoExtension::AddVariable method

Create a CaoVariable object that related to a specific cell within a Database by specifying Field name in Variable name. At that time, you cannot specify a Field name that is designated as a Key name.

This method will fail if any value other than Field name is entered in Variable name.

```
AddVariable
(
    "<Variable name >",                     // Field value
    "<Option>"           // Option character strings
)
```

2.2.5. CaoExtension::get_VariableNames property

Obtain a list of Field name other than the key field of a table.

2.2.6. CaoExtension::Execute method

Delete a record that is related to an object. To delete a record, specify "Delete" in command argument of this method.

Even if a record has been deleted, an object that is related to the deleted object still exist; therefore, you need to delete unnecessary objects from a client program.

This method will fail if other than "Delete" is specified for the command.

```
Execute
(
    "Delete",    // "Delete"(fixed)
)
```

2.2.7. CaoVariable::get_Value method

Obtain a value of cell that is related to a CaoVariable object.

2.2.8. CaoVariable::put_Value method

Set a value into a cell that is related to a CaoVariable object.

2.3. Variable list

There are no DB provider-specific variables.

2.4. Error code

DB provider does not have original error codes. For ORiN2 common errors, refer to ORiN2 programing guide.

3. Sample program

The following shows a sample to obtain/set data by accessing database.

Server name : "DCServer"

Database : Access2000

File path : "C:¥DB.mdb"

Table name Table1

ID	Data
0	Sample
1	Test
2	SampleData

List 3-1 Sample.frm

```
Private eng As CaoEngine
Private ctrl As CaoController
Private ext As CaoExtension
Private var As CaoVariable

Private Sub Form_Load()

    Dim ws As CaoWorkspace
    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    ' Connect to a table
    Set ctrl = ws.AddController("Table1", _
                                "CaoProv.DataBase", _
                                "", _
                                "Provider= Microsoft.Jet.OLEDB.4.0, Data Source =C:¥DB.mdb")

    ' Obtain a record
    Set ext = ctrl.AddExtension("1")

    ' Obtain a cell
    Set var = ext.AddVariable("Data")

End Sub

' Set a variable
Private Sub Command1_Click()
    var = Text1.Text
End Sub

' Obtain a variable
Private Sub Command2_Click()
    Text1.Text = var
End Sub
```

4. Command reference

4.1. Command list

Table 4-1 Command list

Category	Command	Description	
	SQL	Execute SQL.	P. 15
	BeginTrans	Start a transaction.	P. 16
	CommitTrans	Complete a transaction normally (commit).	P. 16
	RollbackTrans	Complete a transaction abnormally (rollback).	P. 16
	Requery	Requery	P. 17

4.2. Command descriptions

SQL

Syntax

object.SQL <SQL>

Argument

<SQL> = VT_BSTR ： SQL sentence

Return value

[RecordsetToArray=True]

<Data> = VT_VARIANT | VT_ARRAY ： Execution result of SQL sentence

[RecordsetToArray=False]

<Data> = VT_DISPATCH ： Execution result of SQL sentence

Description

Execute a SQL sentence specified by <SQL>.

Syntax of SQL sentence is not checked.

The data type of execution result differs depending on the setting of RecordsetToArray option of AddController. TRUE: VARIANT array. FALSE: ADO Recordset object.

BeginTrans

Syntax *object*.BeginTrans

Argument (none)
Return value (none)
Description Start a transaction.

CommitTrans

Syntax *object*.CommitTrans

Argument (none)
Return value (none)
Description Complete a transaction normally (commit).

RollbackTrans

Syntax *object*.RollbackTrns

Argument (none)
Return value (none)
Description Complete a transaction abnormally (rollback).

Requery

Syntax *object*.Requery

Argument (none)

Return value (none)

Description Execute a query again and update Recordset object.

Appendix A. About file name and table name

The error of "0x80040E14" might return when the file name and the Table name include operator (+-*/) etc. in AddController. In this case, enclose the name with the escape symbol of each database.

Example)

When the data source is Excel,CSV, etc. : [test-data.csv]

If the datasource is PostgresSQL: "test-table"

The sample to access the CSV file and to acquire data is shown as follows.

File name : test-data.csv

File path : "C:¥ test-data.csv"

value

id	name	bal
1	1	11
2	2	22
3	3	33
4	4	44

List A-1 Sample.frm

```
Private eng As CaoEngine
Private ctrl As CaoController
Private ext As CaoExtension
Private var As CaoVariable

Private Sub Form_Load()

    Dim ws As CaoWorkspace
    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    ' open csv file
    Set ctrl = ws.AddController("[test-data.csv]", _
                                "CaoProv.DataBase", _
                                "", _
               "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=C:¥; Extended Properties=Text")

    ' get recordset
    Set ext = ctrl.AddExtension("1")

    ' get cell
    Set var = ext.AddVariable("name")

End Sub

' get value
Private Sub Command2_Click()
    Text1.Text = var.value
End Sub
```