

# Database プロバイダ データベースアクセス

Version 1.0.2

## ユーザーズ ガイド

September 15, 2023

【備考】

## 【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-23	初版.
1.0.0.1	2010-02-10	エラーコード追加
1.0.0	2012-07-17	ドキュメントのバージョンルールを変更
	2015-08-03	AddController オプションに RecordsetToArray を追加
	2015-08-05	AddController オプションに SQL を追加
	2015-08-26	Execte のコマンドに BeginTrans, CommitTrans, RollbackTrans を追加
	2015-09-02	Execte のコマンドに Requery を追加
1.0.1	2016-02-19	AddController のオプション文字列に Misc を追加
	2017-06-29	Access2017(*.accdb)の接続例追加
	2017-12-08	PostgreSQL の接続例追加, AddController オプションに@Connected を追加
	2017-12-18	Oracle の接続例修正, Excel2007 以降(*.xlsx, *.xlsm, *.xlsb)の接続例追加
1.0.2	2018-10-30	メモリーク バグ修正
	2018-12-05	付録 A 追記.
	2019-12-26	誤記修正 Access→Access
	2020-12-01	PostgreSQL の説明追加.
	2022-11-21	MySQL の説明追加.
	2023-04-11	誤記修正.
	2023-09-15	誤記修正.

## 【対応機器】

機種	バージョン	注意事項

## 目次

1. はじめに .....	4
2. プロバイダの概要 .....	5
2.1. 概要 .....	5
2.1.1. レコード追加 .....	6
2.1.2. レコード削除 .....	6
2.2. メソッド・プロパティ .....	7
2.2.1. CaoWorkspace::AddController メソッド .....	7
2.2.2. CaoController::AddExtension メソッド .....	11
2.2.3. CaoController::Execute メソッド .....	12
2.2.4. CaoExtension::AddVariable メソッド .....	12
2.2.5. CaoExtension::get_VariableNames プロパティ .....	12
2.2.6. CaoExtension::Execute メソッド .....	13
2.2.7. CaoVariable::get_Value メソッド .....	13
2.2.8. CaoVariable::put_Value メソッド .....	13
2.3. 変数一覧 .....	13
2.4. エラーコード .....	13
3. サンプルプログラム .....	14
4. コマンドリファレンス .....	15
4.1. コマンド一覧 .....	15
4.2. コマンド詳細 .....	15
付録 A. ファイル名・テーブル名について .....	18

## 1. はじめに

本書は、CAO のインタフェースを使用してデータベースにアクセスするデータベースプロバイダ(以下 DB プロバイダ)のユーザーズガイドです。

DB プロバイダは CAO のインタフェースを使用し円滑なデータ操作を行えるように開発されています。データベースの操作は ADO のインタフェースを介して行われており、DB プロバイダは CAO のインタフェースで ADO のインタフェースをラップしながら接続する形で実装されています。また、OLEDB プロバイダに対応したデータベースエンジンは基本的に DB プロバイダでサポートします。

本書は、この DB プロバイダの機能と実装されているメソッドについて説明します。

## 2. プロバイダの概要

### 2.1. 概要

DB プロバイダでは、データベースのテーブル、レコード、セルの値をそれぞれの CaoController, CaoExtension, CaoVariable オブジェクトに対応しています。(表 2-1)これにより、各オブジェクトの名前にテーブル名、キー値、セル名を指定することにより、データベースの特定のセル位置を指定することが可能です。

表 2-1 CAO のクラスとデータベースの対応

クラス	データベース
Controller	テーブル
Extension	レコード
Variable	セル

また、CaoController オブジェクトの Execute メソッドが実装されています。これは ADO.Connection オブジェクトの Execute メソッドに接続されていて、CAO の IF を介して SQL 文の実行も行えるようになっています。

DB プロバイダのファイル形式は DLL(Dynamic Link Library)となっており、その詳細は表 2-2 のようになっています。

表 2-2 DataBase プロバイダ

ファイル名	CaoProvDB.dll
ProgID	CaoProv.DataBase
レジストリ登録 <sup>1</sup>	regsvr32 CaoProvDB.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDB.dll

DB プロバイダには、MDAC (Microsoft Data Access Components) 2.7 が必要です。

<sup>1</sup> ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

### 2.1.1. レコード追加

CaoController::AddExtension メソッドのオプションで“@Create”オプションで 1 以上を指定することで、レコードを追加することが出来ます。このとき拡張ボード名が追加レコードのキー値となります。キー以外のセルには、データは入れないので、値入力が必要のセルがある場合は、レコードの追加は失敗します。

### 2.1.2. レコード削除

CaoExtension::Execute メソッドのコマンドに“Delete”と入力することで、レコードを削除します。レコード削除後に、削除したレコード、セルとつながっているオブジェクトは、自動的に消滅はしません。このようなオブジェクトはクライアントプログラムで削除して下さい。

## 2.2. メソッド・プロパティ

### 2.2.1. CaoWorkspace::AddController メソッド

DB プロバイダでは Controller オブジェクトの生成時に ADO.Connection オブジェクトを作成し、データベースエンジンと接続します。このとき、ADO 接続文字列は DB プロバイダでは特に操作せずに、直接 ADO.Connection オブジェクトの Open メソッドに受け渡します。

以下に AddController メソッドを実行するときの例を示します。

```
AddController
(
    "<コントローラ名>", // テーブル名.
    "CaoProv. DataBase", // プロバイダ名. 固定.
    "<マシン名>", // プロバイダの実行マシン名.
    "<オプション>" // オプション文字列
)
```

以下にオプション文字列に指定するリストを示します。

表 2-3 CaoWorkspace::AddController のオプション文字列

オプション	意味
Provider=<OLEDB プロバイダ名>	開きたいデータベースの OLEDB プロバイダ。(表 2-4)
Data Source=<データソース>	データソース。(表 2-4)
Initial Catalog=<カタログ名>	初期カタログ。(表 2-4)
User ID=<ユーザ ID>	データベースのユーザ ID。(表 2-4)
Password=<パスワード>	上記ユーザのパスワード。(表 2-4)
Extended Properties=<属性>	追加の属性。(表 2-4)
Key[=<キー名>]	キーとして使用するフィールド名 (デフォルト: 指定したテーブルの主キー) DB プロバイダでは複合キー(2 カラム以上にまたがるキー)はサポートされていません
RecordsetToArray[=<True/False>]	CaoController::Execute メソッドの SQL コマンド実行結果形式。 True: VARIANT 配列 False: Recordset オブジェクト (デフォルト)
SQL[=<SQL 文>]	プロバイダ内で使用する Recordset オブジェクトを SQL 文の実行した結果に置換。 (デフォルト: コントローラ名をテーブル名として取得した Recordset オブジェクトを使用)
Misc[=<ADO 接続文字列に追加する文字列>]	ADO 接続文字列に追加したい文字列。 必ず括弧で囲む。 例) Misc=(Database=DB;Port=2345)

@Connected[=<True/False>]	<p>データベースへのデータアクセス方式</p> <p><b>True:</b> 接続型データアクセス(デフォルト) データベースとの接続を維持したまま、各処理を実行します。</p> <p><b>False:</b> 非接続型データアクセス データベースの接続を常時維持することなく、各処理を実行します。</p>
---------------------------	--

表 2-4 データベースの AddController 設定内容一覧

データベース	コントローラ名	指定オプション	入力内容
SQL Server	テーブル名	Provider	“SQLOLEDB.1”
		Data Source	SQL サーバ名
		Initial Catalog	データベース名
		User ID	データベースのユーザ ID
		Password	ユーザのパスワード
Oracle	テーブル名	Provider	“OraOLEDB.Oracle” <sup>2</sup>
		Data Source	Oracle サーバ名
		User ID	データベースのユーザ ID
		Password	ユーザのパスワード
Access97	テーブル名	Provider	“Microsoft.Jet.OLEDB.3.51”
		Data Source	ファイル名(*.mdb)
Access2000	テーブル名	Provider	“Microsoft.Jet.OLEDB.4.0”
		Data Source	ファイル名(*.mdb)
Access2007 以降 (*ACCDB)	テーブル名	Provider	“Microsoft.ACE.OLEDB.12.0”
		Data Source	ファイル名(*.accdb)
Excel2000	[<シート名>\$<範囲>]	Provider	“Microsoft.Jet.OLEDB.4.0”
		Data Source	ファイル名(*.xls)
		Extended Properties	“Excel 8.0” <sup>3</sup>
Excel2007 以降 (*xlsx, *.xlsm, *.xlsb)	[<シート名>\$<範囲>]	Provider	“Microsoft.ACE.OLEDB.12.0”
		Data Source	ファイル名(*.xlsx, *.xlsm, *.xlsb)
		Extended Properties	“Excel 12.0” <sup>4</sup>
CSV ファイル	CSV ファイル名	Provider	“Microsoft.Jet.OLEDB.4.0”
		Data Source	CVS ファイルの存在するフォルダパス
		Extended Properties	“Text”
PostgreSQL <sup>5</sup>	テーブル名	Provider	“MSDASQL”

<sup>2</sup> Oracle Database(Oracle 9.2 以上)へのアクセスで使用できます。それより前の Oracle Database では“MSDAORA”を使用してください。Provider 名以外は“OraOLEDB.Oracle”と指定パラメータは同じです。”

<sup>3</sup> “Excel”と“8.0”の間に空白が入ります。

<sup>4</sup> “Excel”と“12.0”の間に空白が入ります。

<sup>5</sup> テーブル名とフィールド名に大文字が使用されている場合、対象のデータベースサーバーには接続できません。

	接続パラメータ	Misc	Server=<IP アドレス>; Database=<DB 名>; Uid=<ユーザ名>; Pwd=<パスワード>; Driver=PostgreSQL Unicode;
MySQL <sup>5</sup>	テーブル名	Provider	“ODBC”
	接続パラメータ	Misc	Server=<IP アドレス>; Database=<DB 名>; Uid=<ユーザ名>; Pwd=<パスワード>; Driver=MySQL ODBC 5.3 Unicode Driver;

以下に各データベースの AddController の例を示します。

例1 : SQL サーバの場合

サーバ名 : TestServer  
データベース名 : MyDB  
テーブル名 : TestTable  
ユーザ名 : User  
パスワード : Pass

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

例2 : Oracle の場合

サーバ名 : TestServer  
データベース名 : MyDB  
テーブル名 : TestTable  
ユーザ名 : User  
パスワード : Pass

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

例3 : Access2000 の場合

MDB ファイル : C:\¥TestDB.mdb  
テーブル名 : TestTable

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

例4 : Excel2000 の場合

XLS ファイル : C:¥TestDB.xls  
シート名 : Sheet1  
セル範囲 : A1:B10

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

例5 : CSV ファイルの場合

CSV ファイル : C:¥TestDB.csv

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

例6 : PostgreSQL の場合

サーバ名 : TestServer  
データベース名 : MyDB  
テーブル名 : test\_table

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

例7 : MySQL の場合

サーバ名 : TestServer  
データベース名 : MyDB  
テーブル名 : 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 メソッド

このメソッドでは、拡張ボード名としてキー値を指定することでテーブル内のレコードと関連付けられた拡張ボードオブジェクトを作成します。

テーブル内に拡張ボード名と一致するキーがないときは、このメソッドは失敗します。また、テーブル内に拡張ボード名と一致するキーが複数あるときは、レコード番号の若いレコードと関連付けられます。

オプションに@Create オプションを指定することでレコードを新規作成します。拡張ボード名と同じキーが、既に存在するときはそのレコードと関連付けられ、レコードの新規作成はしません。テーブル内に値入力が必要のフィールドが存在するとき、このメソッドは失敗します。

以下に AddExtension メソッドを実行するときの例を示します。

```
AddExtension
(
  "<拡張ボード名>",           // キー値
  "<オプション>"             // オプション文字列
)
```

以下にオプション文字列に指定するリストを示します。

**表 2 CaoController::AddExtension のオプション文字列**

オプション	意味
@Create[=<0~1>]	<ul style="list-style-type: none"> <li>拡張ボード名で指定したレコードがないときこのオプション値に従ってレコードを追加します。</li> <li>0:レコードを追加しない(デフォルト)</li> <li>1:レコードを追加する</li> </ul>

### 2.2.3. CaoController::Execute メソッド

コントローラクラスのコマンドを実行します。  
各コマンドの詳細は 4 を参照してください。

### 2.2.4. CaoExtension::AddVariable メソッド

変数名にフィールド名を指定することで、データベース内の特定のセルと関連付けられた CaoVariable オブジェクトを生成します。このとき、キー名に指定されているフィールド名は指定することができません。

変数名にフィールド名以外の値を入力したときはこのメソッドは失敗します。

```
AddVariable
(
  "<変数名>",                 // フィールド値
  "<オプション>"             // オプション文字列
)
```

### 2.2.5. CaoExtension::get\_VariableNames プロパティ

テーブルのキーと指定されているフィールドを除く、フィールド名の一覧を取得します。

### 2.2.6. CaoExtension::Execute メソッド

このメソッドのコマンド引数に“Delete”を指定することで、オブジェクトと関連付けられているレコードを削除します。

レコードが削除した後も、レコードに関連付けられているオブジェクトは消滅しません。クライアントで不要になったオブジェクトを削除しなければなりません。

コマンドに“Delete”以外を指定したときは、このメソッドは失敗します。

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

### 2.2.7. CaoVariable::get\_Value メソッド

CaoVariable オブジェクトに関連付けられているセルの値を取得します。

### 2.2.8. CaoVariable::put\_Value メソッド

CaoVariable オブジェクトに関連付けられているセルに値を設定します。

## 2.3. 変数一覧

DB プロバイダ固有の変数はありません。

## 2.4. エラーコード

DB プロバイダでは、固有のエラーコードはありません。ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章を参照してください。

### 3. サンプルプログラム

以下にデータベースにアクセスしてデータの取得, 設定をするサンプルを示します.

サーバ名 : “DCServer”  
データベース : Access2000  
ファイルパス : “C:¥DB.mdb”  
テーブル名 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)

    ' テーブルと接続
    Set ctrl = ws.AddController("Table1", _
        "CaoProv. DataBase", _
        "", _
        "Provider= Microsoft. Jet. OLEDB. 4. 0, Data Source =C:¥DB.mdb")

    ' レコードの取得
    Set ext = ctrl.AddExtension("1")

    ' セルの取得
    Set var = ext.AddVariable("Data")

End Sub

' 変数の設定
Private Sub Command1_Click()
    var.value = Text1.Text
End Sub

' 変数の取得
Private Sub Command2_Click()
    Text1.Text = var.value
End Sub
```

## 4. コマンドリファレンス

### 4.1. コマンド一覧

表 4-1 コマンド一覧

カテゴリ	コマンド	機能	
	SQL	SQL 実行	P. 15
	BeginTrans	トランザクション開始	P. 16
	CommitTrans	トランザクション正常終了	P. 16
	RollbackTrans	トランザクション異常終了	P. 16
	Requery	クエリ再実行	P. 17

### 4.2. コマンド詳細

## SQL

### 構文

*object*. SQL <SQL>

### 引数

<SQL> = VT\_BSTR : SQL 文

### 戻り値

[RecordsetToArray=True]

<Data> = VT\_VARIANT | VT\_ARRAY : SQL 文実行結果

(<LabelList>, <Record1>, <Record2>, ...)

<LabelList> = VT\_BSTR | VT\_ARRAY : 列名リスト (<Label1>, <Label2>, ...)

<Label n> = VT\_BSTR : 列名

<Record n> = VT\_VARIANT | VT\_ARRAY : レコードデータリスト

(<Data1>, <Data2>, ...)

<Data n> = VT\_VARIANT : データ。

データの列名は同じインデックス番号の列名になります。

[RecordsetToArray=False]

<Data> = VT\_DISPATCH : SQL 文実行結果

### 説明

<SQL>で指定した SQL 文を実行します。

SQL の文法チェックは行いません。

実行結果は AddController の RecordsetToArray オプションの指定が True の場合は VARIANT 配列で、False の場合は ADO Recordset オブジェクトで得られます。

---

## BeginTrans

---

構文	<code>object.BeginTrans</code>
引数	なし
戻り値	なし
説明	トランザクションを開始します。

---

## CommitTrans

---

構文	<code>object.CommitTrans</code>
引数	なし
戻り値	なし
説明	トランザクションを正常終了(コミット)します。

---

## RollbackTrans

---

構文	<code>object.RollbackTrns</code>
引数	なし
戻り値	なし
説明	トランザクションを異常終了(ロールバック)します。

---

# Requery

---

構文	<code>object.Requery</code>
引数	なし
戻り値	なし
説明	クエリを再実行し Recordset オブジェクトを更新します。

## 付録 A. ファイル名・テーブル名について

AddController 時にファイル名や Table 名に演算子(+\*/)等が入っていると「0x80040E14」のエラーが返ることがあります。その際には各データベースのエスケープ記号で名前を囲って指定してください。

例)

データソースが Excel, CSV などの場合: [test-data.csv]

データソースが PostgreSQL の場合: “test-table”

以下に CSV ファイルにアクセスしてデータの取得をするサンプルを示します。

ファイル名 : test-data.csv  
 ファイルパス : “C:¥ test-data.csv”  
 ファイル

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)

    ' テーブルと接続
    Set ctrl = ws.AddController("[test-data.csv]", _
        "CaoProv.DataBase", _
        "Provider=Microsoft.Jet.OLEDB.4.0, Data Source=C:¥, Extended Properties=Text")

    ' レコードの取得
    Set ext = ctrl.AddExtension("1")

    ' セルの取得
    Set var = ext.AddVariable("name")

End Sub

' 変数の取得
Private Sub Command2_Click()
    Text1.Text = var.value
End Sub
```