

## ■ ZEBRADLL.NET 仕様書

### 概要

基本的にこの DLL はデータベースアプリケーションを簡単に作成するために作成したライブラリである。いちいち、データベースへの接続文字列とかを気にすることなく決まり文句の様にデータベースとアプリケーションを接続してしまう事を想定した。

また、.NET 仕様に切り替えたことで過去作成していた vb6.0 の時の DLL を手直しして ADO.NET に完全に移行したので Windows 7 でも問題無く動作する。

この DLL の特徴は、MDB、ACCDB (Access2007)、MySQL、MS SQLServwer の 4 つのデータベースを視野にいて作成しており、いずれのバックエンドでも表向きはほぼ同様に扱える事を念頭に置いている。

ただし、SQL の方言に気をつけなければ Access などでの wild card は「\*」アスタリスクでいけるが SQL を発行する場合「%」に置き換えなければエラーとなるなどスタンダードな SQL 文を作成することで、アップサイジングなどの時にスムーズに移行出来様にしている。

当初は Access でネットワーク共有していたが台数が増えたので SQLServwr へ切替る時にも少々の工数で作業を終えることが出来る。

しかし、これは内部的な事で、エンドユーザは知るよしもなく、正規の料金で請求することも可能だ。

話は余談にそれだが、この DLL を使えば比較的容易に DB アプリが構築出来るのではないと思う。

### ■接続の決まり文句 (サンプルソースコード解説)

VB.NET2008 を対象に説明していく

ZEBRADLL を使える様に宣言する ZDLL の部分は自分の好きな名前にすれば良いでしょう。

```
Public ZDLL As New zebra_net.comFnc
```

次に接続するコネクションの宣言でいずれか一つ、ターゲットにしている DB のみで良い

Access :           Public CURRENTDB As System.Data.OleDb.OleDbConnection

MySQL :           Public MyConn As MySql.Data.MySqlClient.MySqlConnection

SQLServer :       Public MsConn As System.Data.SqlClient.SqlConnection

### 参照設定

MySQL.Data   MySQL を扱う場合必要

System.data   MDB、ACCDB、MSSQL はこれが必要

ZEBRADLL.net 自体の参照も忘れずに

変数の宣言の決まり文句はこれだ！

```
Dim errMsg As String = ""
```

```
Dim errNmb As Long = False
```

```
Dim strSQL As String = ""
```

```
Dim ADORS As System.Data.OleDb.OleDbDataReader = Nothing
```

```
Dim MsADO As System.Data.SqlClient.SqlDataReader = Nothing
```

```
Dim MyADO As MySql.Data.MySqlClient.MySqlDataReader = Nothing
```

下の設定は ZEBRADLL でログを残す為に ZEBRADLL にパスを渡している

```
With ZDLL
```

```
  .appPth = Application.StartupPath
```

```
  Appname = Replace(Application.ExecutablePath, Application.StartupPath + "\¥", "")
```

```
  .appNam = Appname.Substring(0, Len(Appname) - 4)
```

```
  Call .iniLog()
```

```
End With
```

接続 ソースコードを見ればわかるとおり、サンプルは DB\_MOD という変数で接続先を切り替える様にしているが通常いずれか一つを利用すればよい。

```
'DB 接続
Select Case DB_MOD
    Case "MDB" 'MDB
        If ZDLL.conectMDB (DBPW, aPPpath & DBP, CURRENTDB, errMsg, errNmb) = False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
        strSQL = "SELECT * FROM 26KYOUTO;"
    Case "MySQL" 'MySQL
        If ZDLL.conectMySQL (MySVR, MyDBP, MyDBID, MyDBPW, MyConn, errMsg, errNmb) =
False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
        strSQL = "SELECT * FROM テーブル名;"
    Case "MsSQL" 'MSSql
        If ZDLL.conectMsSQL (MsSVR, MsCAT, MsDBID, MsDBPW, MsConn, errMsg, errNmb) =
False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
        strSQL = "SELECT * FROM テーブル名;"
    Case "ACCDB" 'ACCDB
        If ZDLL.conectACCDB (DBPW, aPPpath & DBP1, CURRENTDB, errMsg, errNmb) = False
Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
        strSQL = "SELECT * FROM 26KYOUTO;"
End Select
```

※サンプルは接続と同時に SQL 文字列もセットしているだけでここは無くても良い

## ■レコードセット（データリーダー）の取得

ADO での接続はレコードセットが取得でき、ADORS.FILED ("○○") = ××でアップデートするとそれで書き込めたのだがレコードリーダー接続 (ADO.NET) になるとそのやり方が出来なくなり、完全な SQL を書かなければならなくなってしまった。

ただし、SELECT、INSERT、UPDATE、DELETE を正しく書かなければならないので移植する場合には都合がよいので今は、完全な SQL 文を書いて Execute する方式に切り替えた。

```
'レコードセット取得
Select Case DB_MOD
    Case "MDB", "ACCDB"
        If ZDLL.getRecSet(CURRENTDB, strSQL, ADORS, errMsg, errNmb) = False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
    Case "MySQL"
        If ZDLL.getMyRecSet(MyConn, strSQL, MyADO, errMsg, errNmb) = False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
    Case "MsSQL"
        'MS (SQLServer の場合レコードセットはレコードリーダーに変わった.net)
        If ZDLL.getDataReader(MsConn, strSQL, MsADO, errMsg, errNmb) = False Then
            MsgBox( errNmb & vbCr & errMsg, MsgBoxStyle.Information,
Application.ProductName.ToString())
            Exit Sub
        End If
End Select
```

## ■ DataGridView に取得したデータリーダーを一気に貼り付ける場合下記のデータテーブルをセットしておく

```
Dim dt As New System.Data.DataTable
```

で、一端データテーブルにロードしてグリッドのデータソースとして設定してやる。

```
Select Case DB_MOD
    Case "MDB", "ACCDB"
        dt.Load(ADORS, LoadOption.OverwriteChanges)
        DataGridView1.DataSource = dt
    Case "MySQL"
        dt.Load(MyADO, LoadOption.OverwriteChanges)
        DataGridView1.DataSource = dt
    Case "MsSQL"
        dt.Load(MsADO, LoadOption.OverwriteChanges)
        DataGridView1.DataSource = dt
End Select
```

## ■接続語の決まり文句

この上に SQL 文字列をのせる

```
StrSQL = " Select "  
StrSQL += " ***** "  
StrSQL += " ***** "  
StrSQL += " ***** "  
StrSQL += " ***** "  
StrSQL += " ***** "
```

と後で見てわかるように SQL を改行して書いて文字列にイれて SQL を発行するだけ

```
If ZDLL.getRecSet(CURRENTDB, strSQL, ADORS, errMsg, errNmb) = False Then  
    MsgBox(errNmb & vbCr & errMsg, MsgBoxStyle.Information, Application.ProductName.ToString())  
    Exit Sub  
End If
```

このあと ADORS.Close() を忘れると SQLServer などではエラーが出る。MDB は結構横着でクローズしなくてもエラーは無視して実行出来る。(イミディエイドには例外が一杯でてくるけど)

フィールドの値を取得するのは ADORS("フィールド名") で取得出来る (簡単)  
ただし、取得する前に ADORS.Read() を実行しておかないと Null になってしまうので注意

### 手順

決まり文句で ADORS にレコードリーダーを取得する (中身を入れる)  
ADORS.Read() をコールする。  
ADORS (" フィールド名") で値を取得したり色々  
ADORS.Close() 閉じるのを忘れずに

細かい事はレコードリーダーの取り扱いを調べればわかるでしょう。  
レコードリーダーについてはここではこれ以上触れませんが簡単にレコードリーダーと言う ADO.NET の流儀にのっとして DB アプリを作成することが出来る様になります。

VB6 ユーザにはありがたい機能です。自分で言うのも何ですが。

基本的にフリーで提供しているのですが何かわからないことは下記まで。

tamayan@zebrasoft.co.jp

<http://zebrasoft.co.jp/blog/>でブログ書いてるので覗いてやってください。

何となく一人で色々ソフト作って生きてます。

MZ2000 時代の生き残りです。グリーンモニターってわかるかなあ。

PC8001 とか 98 とかマウスなんか無かったもんなあ。今の VisualStudio はプログラム知らなくても誰でもプログラム作れる様になったから凄い進歩だな。

昔は画面に円を書くにもサインコサインで書いたもんだけどなあ.....