

Visual Studio 2008 C#で FoxProを運用する

Object Oriented XBASE Forum :Nobuyuki Ichioka

掲載開始2009年1月14日

更新日 2009年1月24日

FoxProで作成したマルチスレッドのCOM (DLL)が.NET上でも動作することが知られていますが、COM内のマクロとEXECSCRIPT ()を.NET側から呼び出すことで、xBASEプログラムを実行できることが判明いたしましたので、その手順をご説明申し上げます。

この手順を実行利用するにあたり、いくつかの問題点がありますので以下に列挙いたします。

- EULA (エンドユーザライセンス)に抵触するか不明。
- エラー処理を実施していません。
- 作成DLLをレジスリーする必要がある。
- 検証したパソコンはVFP9.0がインストールしてあり、かつVS2008がインストールしてあるパソコンでのみ行っており、ランタイム (FORBS9等と)のみのインストールパソコンでの動作について検証していない。
- 他のパソコンにC#で作成したアプリ (EXE)とこのDLLをインストールして作動を確かめていない

※上記につきまして試運転を実施するパソコン環境に十分ご注意ください。

使用環境

FoxPro Ver9.0 SP2

Visual Studio 2008 Professional Edition

※C#を利用してご説明してあります。

I) FoxProでDLLを作成する

FoxProを立ち上げ(たぶんVer7.0でも可 Ver6.0は不可?)空のプロジェクトを作成します。このとき、ウィザードを使わないよう **[File]-[New]-OProject-□New file** で作成します。また、プロジェクトのフォルダー位置(深さ)と名称を短めにつけることをお勧めします。これはVistaの場合、DLLの登録にあたりregsvr32をコマンドプロンプト窓から実施する必要があります。ディレクトリーパスの階層を下げてくださいしんどくなるのを防ぐためです。

※ちなみにサンプルは c:¥VFPCOM¥VFP_COM.prjにしました。

次に、Project Managerの**[Code]-[Program]-[New]** を選択しProgramエディターを開けます。デフォルトでProgram1と付いたウインドウが開きます。

次項にそのウインドウに記入するプログラムを表示します。

Programとして記述するのは下記のとおりです。

```
DEFINE CLASS VisualFoxProForDotNet AS Session OLEPUBLIC
cStartPath = ""
glcRetVal = ""
glcMacro = ""

PROCEDURE INIT
    SET RESOURCE OFF
    SET EXCLUSIVE OFF
    SET REPROCESS TO 2 SECONDS
    SET CPDIALOG OFF
    SET EXACT OFF
    SET SAFETY OFF
    This.cStartPath = ADDBS(JUSTPATH(Application.ServerName))
    SET PATH TO (This.cStartPath)
ENDPROC

PROCEDURE SetVFPPCommand(sCommand AS String) AS String
*HELPSTRING ""
    glcRetVal = ""
    glcMacro = sCommand
    &glcMacro.
RETURN glcRetVal
ENDPROC

PROCEDURE SetVFPPProgram(sProgram AS String) AS String
*HELPSTRING ""
    glcRetVal = ""
    glcMacro = sProgram
    EXECSCRIPT(glcMacro)
RETURN glcRetVal
ENDPROC

ENDDDEFINE
```

上記のプログラムを書いた後にファイル名を(仮に)vfpfdnと登録(c:\¥vfpcom¥vfpfdn.prg)してください。

あとは、コンパイルするだけです。

Project Managerダイアログの [Build] ボタンを押し、Build Optionダイアログで、
Multi threaded COM server [dll]を選択、Recompil All Filesと
Display Errorsにチェックを入れます。OKボタンを押します。

DLLファイル名称は、(仮に) **Vfp4DotNet.dll** としてください。Vistaの場合DLLの登録が出来なかったとエラーが表示されますが、これは正常です。

エクスプローラで c:\¥vfpcomに vfp4dotnet.dll tlb VBRが作成されているのが確認できると思います。

Vistaの場合、DLLを手動で登録する必要があります。

[スタート]-[プログラム]-[アクセサリ]でコマンドプロンプトの上でマウスを右クリックします。メニューから[管理者として実行]を選択します。

コマンドプロンプトで

```
>CD c:¥¥vfpcom
```

```
>regsvr32 vfp4dotnet.dll
```

とタイプします。

これでDLLの登録が完了します。

SetVFPCCommand(“コマンド文字列”)は、コマンド文字列にセットした単一のコマンドラインをマクロで実行させます。必要な場合は、`gIcRetVal`に戻り値をもどすよう書きます。

C#側での記述例

```
VisualFoxProForDotNet VFPPDN = new VisualFoxProForDotNet();  
...  
VFPPDN.SetVFPCCommand("use c:¥¥vfpcom¥¥table1 in 0");  
VFPPDN.SetVFPCCommand("set filter to IDCODE = '1'");  
this.textBox1.Text = VFPPDN.SetVFPCCommand("CURSORTOXML('table1', 'gIcRetVal', 0, 0, 0, '1')");  
VFPPDN.SetVFPCCommand("use in table1");  
...
```

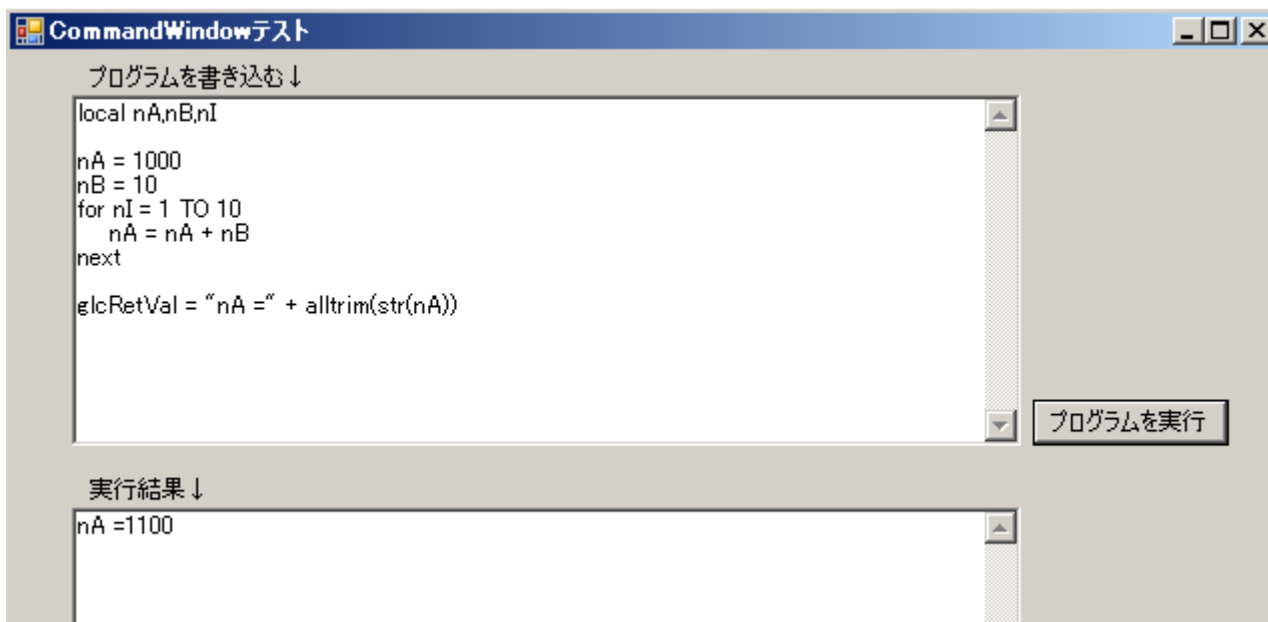
SetVFPPProgram(“プログラム文字列”)は、プログラムを記述した文字列を実行させます。SetVFPCCommandに比し、複雑なプログラムを記述できます。結果は、`gIcRetVal`に文字列で戻すようにします。

C#側での記述例

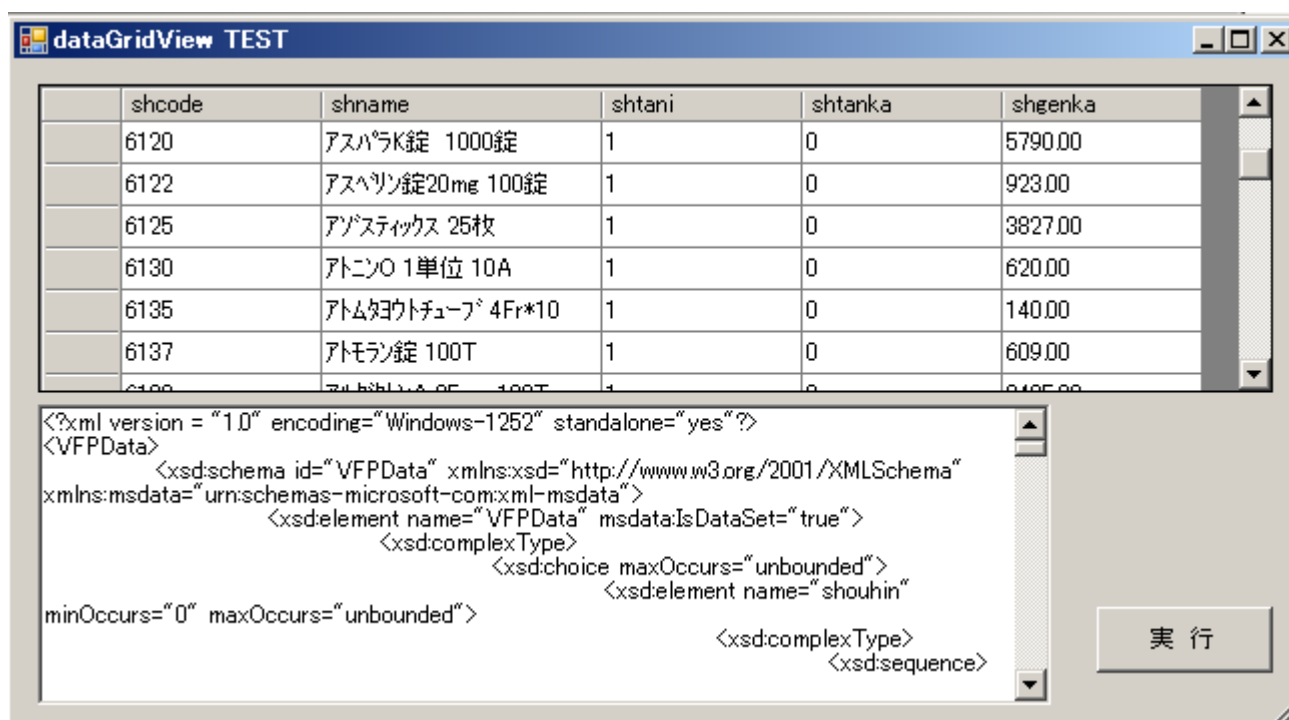
```
VisualFoxProForDotNet VFPPDN = new VisualFoxProForDotNet();  
private DataSet dbc1;  
BindingSource binsou = new BindingSource();  
...  
private void button1_Click(object sender, EventArgs e)  
{  
    string cProg;  
    string cXML;  
    dbc1 = new DataSet();  
    cProg = "use c:¥¥vfpcom¥¥table1 in 0 ¥r¥n"  
        + "SET FILTER TO IDCODE = '1' ¥r¥n"  
        + "CURSORTOXML('table1', 'gIcRetVal', 0, 0, 0, '1') ¥r¥n"  
        + "use in table1 ¥r¥n";  
    cXML = VFPPDN.SetVFPPProgram(cProg);  
    textBox1.Text = cXML; // テキストボックスにXMLの原文を表示  
    dbc1.ReadXml(new StringReader(cXML));  
    binsou.DataSource = dbc1.Tables[0].DefaultView; //グリッドに表示  
    dataGridView1.DataSource = binsou;  
}
```

C#での実行例を画像で示します。

```
private void button1_Click(object sender, EventArgs e)
{
    string cText;
    string cRetVal;
    cText = textBox1.Text;
    cRetVal = VFPPDN.SetVFPPProgram(cText);
    textBox2.Text = cRetVal;
}
```



前項のプログラムでdataGridViewにXMLで受けたテーブルデータを表示した例



肝心なことが前後しましたが、C#へのDLLの登録はソリューションエクスプローラの[参照設定]を右クリックし[参照の追加]を選択します。そして参照の追加ダイアログの[COM]タブを選択しリストからvfp4dotne Type Libraryを選択します。

C#プログラムヘッダーで

```
using vfp4dotnet;  
using System.IO;
```

2行を追加します。

TextBoxに記述した内容でプログラムを実行する場合、c:¥¥vfpcom¥¥ というような¥のエスケープシーケンスを考慮する必要はありません。c:¥vfpcom¥で結構です。¥¥とするのはコンパイル時ソースリストに書かれたファイルパスのみの注意事項です。

Unicodeは考慮する必要がなく replace fieldname with '文字列' はそのままshift-JISで登録されます。・・検証が十分でないのでたぶん(^_^;)

C#の場合 プログラム文字列を構成する場合、文字列はダブルクォートで一番外側をくくります。よってプログラム内はシングルクォートの制約があります。またプログラム行の改行表現は ¥r¥nで行ってください。

COMのサンプルをご用意いたしました。

FoxPro Ver9.0用をお持ちでない場合、FORBS9をインストールする必要があります。なを「茶殻」をご利用いただく場合、ACTIVEDOCUMENTSテクノロジーを利用するために、FoxPro Ver7.0または、FORBS7のインストールが必要になります。

FoxPro Ver9.0をお持ちの場合、またはFORBS9をインストールした後にFoxPro Ver7.0またはFORBS7のインストールを行ってください。FoxProVer7.0⇒FoxPro9.0の順でインストールを行うとACTIVEDOCUMENTSの起動が出来なくなります。繰り返しますがVer9.0⇒Ver7.0の順でインストールを実施します。

製作者は、使用者がこれらのアプリケーションによって発生する如何なる損害に対してもその責任を負いません。

あらかじめご理解のうえご利用ください。

インストールは専用のサイトをご用意いたしました。サイトに入るためには、クローラー対策のため、ユーザー名とパスワードが必要となります。ユーザー名は VFPCOM パスワードは FORDOTNET です。サイトはSilverLight2で作成したかったのですが、技量不足で今回はFLASHで作成してあります。FLASHの動作するブラウザー環境でアクセス願います。

http://www2u.biglobe.ne.jp/~objxbase/vfptoolkitnet/VFP4DOTNET_DL_PORT.html

上記アドレスのWEBサイトにてご利用条件をご確認の上ダウンロードを行ってください。

参考にしたWEBページ：

http://foxcentral.net/microsoft/NETforVFPDevelopers_Chapter15.htm