

- 「ActiveXDll」の作成方法＝「VBAの参照設定で使えるDll」の作成（COMタイプ?のDll。Declareステートメントでは呼びだせないモノ。Declareステートメントで呼び出せるDllは、VC++で作ります。）

※昔メモしたやつなので、呼び出しなどで理解し間違えている箇所があるかもしれません。

ここでのDllはCOMタイプ(?)のDLLなのか、よくわからないんですが、とにかくDeclareでは呼び出せないらしいです。よってクエリでも使えません。＝当然ピボットもダメです。

ただし、クエリではユーザー定義関数でラップすれば使えます。

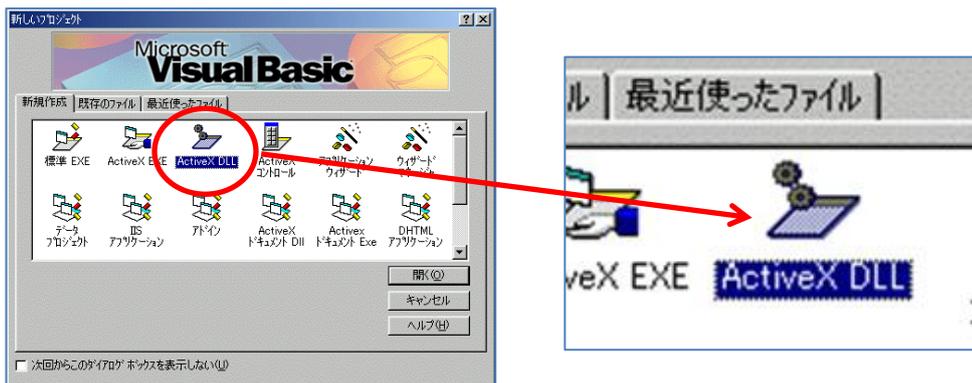
また、「Instancing」の設定を「6-GlobalMultiUse」に変更してからDllを生成すれば、VBE内では（参照設定後）直接呼び出せます。

★DLLの作り方 その1

●DLLの作り方

(01)新規プロジェクトファイルの作成

VB6を起動して「ActiveXDll」をダブルクリック。



コードウィンドウが出るが、まず先にプロジェクト名とクラス名を決める。

プロジェクトエクスプローラ等にて

プロジェクト名：comtest01

クラス名：comtestClass1

などに変更する。

プロジェクト名やクラス名はVBAから呼び出す時に使うので必ず意味のわかる名前にすること。

名前が決まったらいったん保存する。

※SETやNEWしなくても呼び出せるようにするにはここで「★DLLの作り方 その2」を参照。

(02)以下のコードを貼り付け

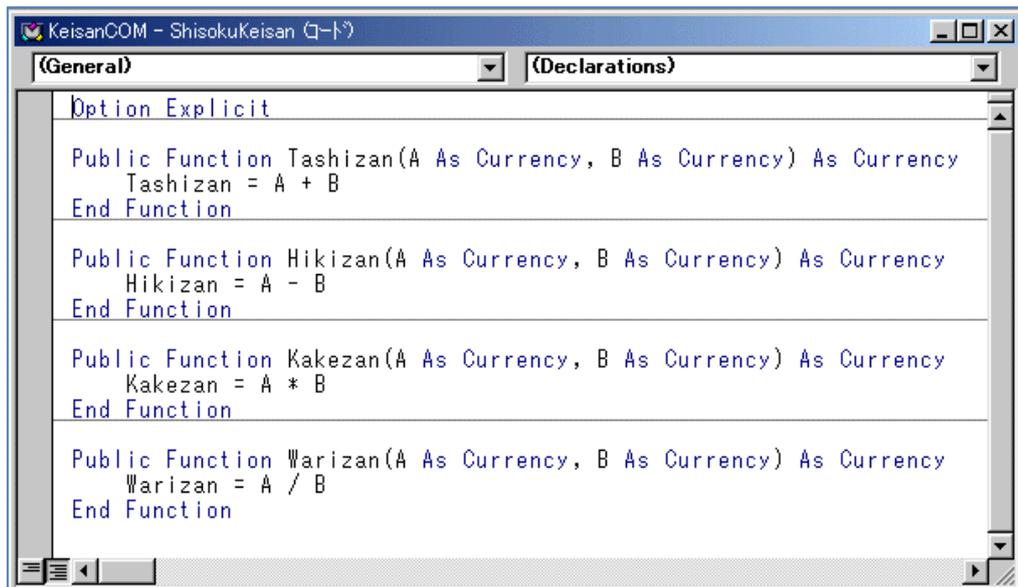
Option Explicit

```
Public Function Tashizan(A As Currency, B As Currency) As Currency
    Tashizan = A + B
End Function
```

```
Public Function Hikizan(A As Currency, B As Currency) As Currency
    Hikizan = A - B
End Function
```

```
Public Function Kakezan(A As Currency, B As Currency) As Currency
    Kakezan = A * B
End Function
```

```
Public Function Warizan(A As Currency, B As Currency) As Currency
    Warizan = A / B
End Function
```



(03)プロジェクトを保存

デフォルトの場所でいい。

C:\Program Files\Microsoft Visual Studio\VB98

comtest01.dll がここで生成されたことを確認する。

(04)DLL の生成

ファイル→××× (プロジェクト名) .dll の作成 を押して、デフォルトの場所に保存 (生成) この場合、「comtest01.dll」ができる。

●VBA からの呼び出し方

comtest01.dll に参照設定をする。
この場合、C:¥Program Files¥Microsoft Visual Studio¥VB98 を指定して普通に dll を参照設定すればよいだけ。

例 01：インテリセンスが使えなくてもよい場合

```
Sub test01()  
  
    Dim a As Object  
    Set a = CreateObject("comtest01.comtestClass1")  
  
    Debug.Print a.Tashizan(1, 2)  
    Debug.Print a.Hikizan(1, 2)  
  
End Sub
```

例 02：インテリセンスを使えるようにしたい場合

```
Sub test02()  
  
    Dim a As New comtest01.comtestClass1  
  
    Debug.Print a.Tashizan(1, 2)  
    Debug.Print a.Hikizan(1, 2)  
  
End Sub
```

例 03：グローバルに使えるようにしたい場合

グローバル宣言部で Set を使うと
「コンパイルエラー プロシージャの外では無効です」エラーになるので
Dim A As New comtest01.comtestClass1 で定義します。

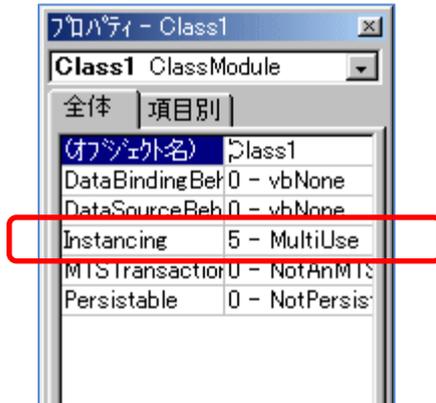
```
Option Explicit  
Dim A As New comtest01.comtestClass1  
-----  
Sub test01()  
  
    Debug.Print A.Tashizan(1, 2)  
    Debug.Print A.Hikizan(1, 2)  
  
End Sub
```

●VBS からの呼び出し方

```
Dim a  
Set a = Wscript.CreateObject("comtest01.comtestClass1")  
  
msgbox a.Tashizan(1, 2)  
msgbox a.Hikizan(1, 2)
```

★DLL の作り方 その2

その1で作るときに、プロパティウィンドウの「Instancing」を「6-GlobalMultiUse」に変更してから DLL を生成すると、
「その1」のように SET や NEW しなくても普通に Call できるようになる。
しかし、クエリでは直接呼び出せない。



★DLL の中身について

DLL では

```
Public Function fMyPath() As String
    'プログラム終了まで MyPath の内容を保持
    Static MyPath As String
    '途中でディレクトリが変更されても起動ディレクトリを確保
    If Len(MyPath) = 0 & Then
        MyPath = App.Path 'ディレクトリを取得
        'ルートディレクトリかの判断
        If Right$(MyPath, 1) <> "\" Then
            MyPath = MyPath & "\"
        End If
    End If
    fMyPath = MyPath
End Function
```

```
Public Function LenA(mozi As String) As Integer
    LenA = LenB(StrConv(mozi, vbFromUnicode))
End Function
```

というように引数も関数名もばらばらのものを格納できる。

Excel でも Access でも使える共有の処理などを DLL 化するとすっきりとして良いのかもしれない。

文字操作（住所、カナ漢字の区別、文字数など各種のチェックなど）
数値操作（年齢、
日付操作（締日算出、締め期間の開始日、終了日などの算出）

といったことなど。