

VBでJ-DLL Serverを使う

1 はじめに

J言語への郷愁は昔、APL言語を少し齧った経験があったが当時はメインフレームの専用TSS端末からの利用であったことが原因である。

しかし、一般企業ではTSS利用環境がそれほど裕福ではないこと、メモリサイズ、CPUの処理速度などの問題などから、実用には使い勝手が良くなかったため、数値計算を含む科学計算処理にはFortranが主流だった。

1986年に、社命で台湾の現地法人に赴任し、1994年に帰国する間に、世の中はすっかりパソコンに席卷され、パソコンで動作するというJ言語が出現しており鈴木義一郎先生がすっかりはまっておられた。

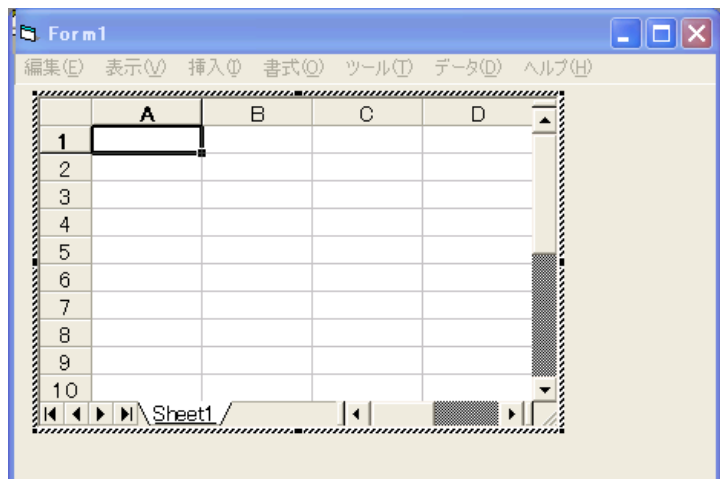
2006年11月より、鈴木先生の紹介でJ言語研究会に参加させて頂き、様々な研究発表を拝見して、昔のAPL言語と比較してJ言語の進化の度合いが計り知れないほど大きいことを知った。つまり、システム開発者にとっては、数値解析プログラムの研究開発に要する時間を大幅に短縮できるし、開発されたプログラムの利用者にとっては、Jのプロセッサがインストールされていなくても実行できる環境が整っているため、これらを十分に普及すればJ言語は一般社会に貢献できる、と考えた。

そこで、これまで2～3回の研究会での皆さんの発表論文を参考に、JをVBに組み込み、Jそのものが動作したり、バックグラウンドでJが動作する仕組みを実験してみた。

2 VB開発環境

現在、マイクロソフト社のVB (Visual Basic) は、.Net (ドットネット) プラットホームが最新であるが、筆者の年齢では .Net を導入してもその能力を使いきれぬわけではなく、まだ十分に使えるとの判断からVB 6.0での環境で実験を試みた。

VBの長所は開発者にとってGUIの組み込みが容易で、開発所要時間やバグを少なくする効用もある。



3 VBの構造

VBは、いわゆるOLE機能を簡単に取り入れることが可能な言語プロセッサである。OLEは、複数のアプリケーションで作成したアイテム、つまり "オブジェクト"

を収容するための仕組みである。OLE は当初、オブジェクトのリンクと埋め込み (Object Linking and Embedding) の略称であった。現在では、単に OLE と呼ばれている。OLE のうち、リンクと埋め込みに関連しない部分は、現在は Active テクノロジー の一部となっている (通称 Active X) 。図は、VB のユーザ画面に EXCEL の OLE オブジェクトを埋め込んだ例である。

VB は、ユーザインターフェイスである画面を基本としてプログラムコードが存在する。もちろん画面を伴わないコードも存在する。プログラムは、基本的にユーザが画面上で何らかのアクション (VB ではイベントと称している) を発生させることにより動作する。基本的には、EXCEL の VBA と同じである。

4 J-DLLserver の組み込みと実験

J-DLL の組み込み方法は、J-SoftWare 社の WebSite

「http://jsoftware.com/help/user/j_ole_auto_server.htm」または、J 言語のヘルプより、「system\extras\help\user\j_ole_auto_server.htm」を参照して頂きたい。

もっとも簡単な方法は、志村氏の「J for WIN9x/NT 入門・EXCEL とのリンク他 (2003 年 8 月 15 日)」の 12 ページを参考にするのが早い。

また、VB プロセッサをお持ちの方は、以下のルートディレクトリから、VB のテストプログラムを入手できる。このルートは、J をインストールした場所で、

「……\j601\system\examples\ole\vb」である。このテストプログラムは、3 種類収録されている。これらのプログラムからの DLL 関数からの戻し値の一部が文字化けする現象があるが、このような現象は英語版の DLL やコントロールを日本語版 OS で使用した場合に多いことが判っている。

筆者は、J-DLLserver を対象にしているが、テストプログラムの対象範囲は J-EXEserver にも及んでいる。J-DLLserver や J-EXEserver は、VB 言語や C 言語の Active X として動作するが、これらの Server に設定されているプロパティやイベントの詳細仕様が不明なため試行錯誤でプログラミングしているのが現状である。

Break interrupt J execution

Clear erases all definitions in J

Do execute a J sentence

ErrorText/ErrorTextM

get error text (run after a J error)

Get/GetB/GetM get the value of a J variable

IsBusy returns 0 if J is ready to execute, else an error code

Log display (1) or discard (0) the J EXE session log

Quit causes J EXE server to close when last object is released

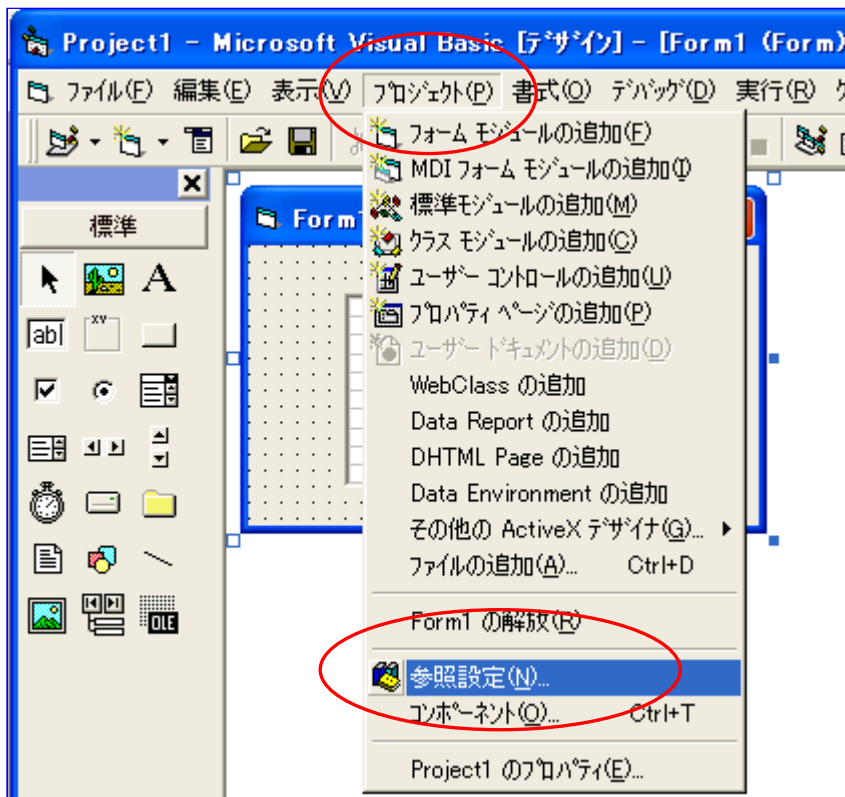
Set/SetB/SetM set a value to a J variable

Show show (1) or hide (0) the J EXE server

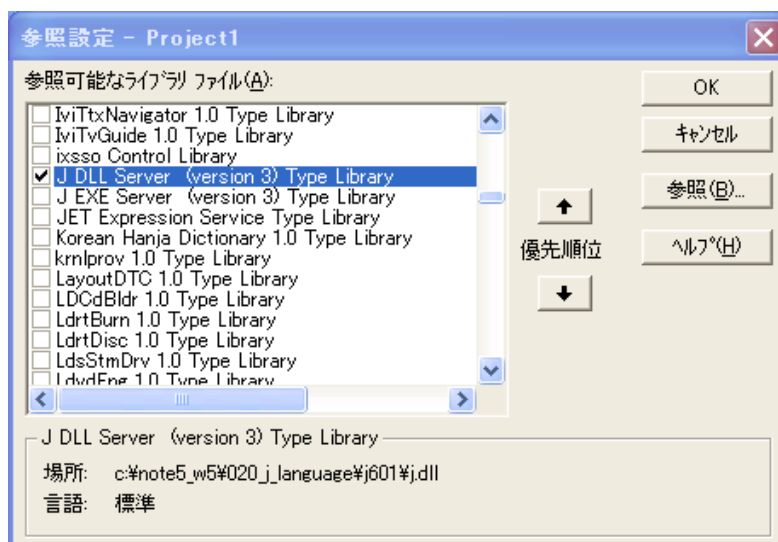
Transpose return array data transposed

J 以外の言語やアプリケーションから、J-DLLServer を利用するには一定の法則がある。志村氏の報告にもあるとおり、初めに J-DLLServer の宣言が必要である。

4. 1. VB での DLL 参照宣言



☆VB プロセッサ (VB エディタ) の「プロジェクト」→「参照設定」をクリック



☆参照設定画面より「J DLL Server(...)Type Library」を探してレ印をつけて参照設定は終了し、VB の中で「J DLL Server」が使用可能となる。

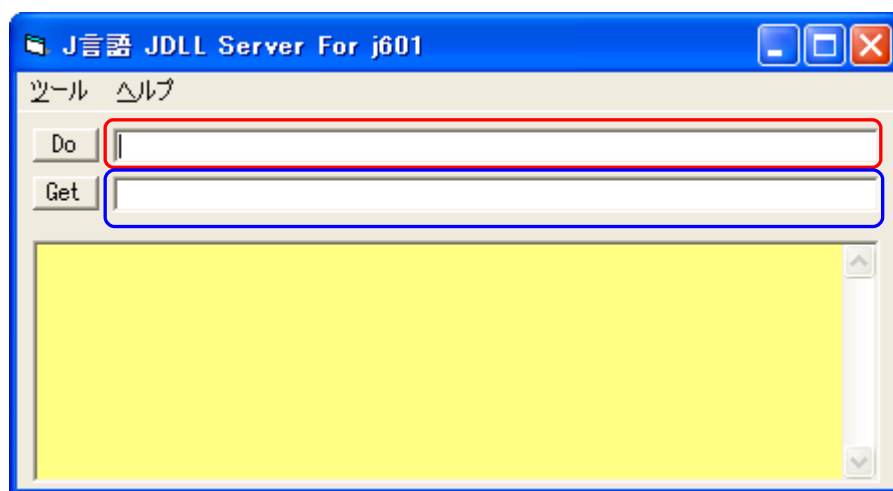
4. 2. VBによるJ関連のコーディング

J-DLLserverの宣言

```
Public Js As Object
Public result As Valiant
Private sub SetJDLL ()
    Set Js = New JDDLserver
End Sub
```

Jコマンドの入力

VBではユーザー・インターフェースのひとつ「TextBox」(赤枠)を使って必要項目を入力する。



TextBoxに入力されたデータは、Textプロパティによって取り出すことができる。TextBoxには名前がつけられるが、ここでは「DoText」(赤枠)と名づけたので、Textプロパティは

```
DoText.Text
```

で取得できる。

Textに格納されたJコマンドを実行するには、J-DLLのJ実行コマンドを呼び出す必要がある。それは、

```
result = Js. Do (DoText.Text)
```

で計算が終了である。ただし、計算結果は出力されません。

DoText.Textの内容が、たとえば

```
DoText.Text = "A=:i.2 2"
```

のときは、「i.2 2」の結果がJ-DDLが確保した「A」という名前の変数に保存された状態になっているので、VBでこの結果を使用することが出来る[↓]。



もし、計算結果を表示したい場合は、「GetText」（青枠）のTextBoxにJコマンドを入力して「Enter」キーを叩く[↓]。



もちろん、計算結果も残し、結果も表示することも出来る[1]。



そして、ここではAに出来た2 X 2の行列の逆行列をJ言語で求めている。その結果は、画面に表示されると同時にBに保存されている[1]。



5 VBプログラム全体

VBプログラムは、先に述べた J601 ファイルの「j601\system\examples\ole\vb」にある VB コード「jsrv1.vbp」を参考に、筆者が独断と偏見で書き変えたものである。この作業の最大の目的は、

J 言語の普及を図るには、J を扱いやすくすることが最大の解決策と考えたからである。普段 J を使い慣れている研究者やそれに準ずる方々にとっては、とても使い易く簡単な言語だと思いがちだが、J 言語を道具としてのみ利用して問題解決を志す人には、J の高度な理論を吸収している暇がない方もいる。そこで、まず J 言語プロセッサをインストールしなくても、簡単に J が使える環境を用意することが肝心ではないかと考えた次第である。その環境を利用している間に、J の素晴らしさ、J でなければ何倍もの時間を消費することなどを実感してもらえれば、必然的に利用者が増えるのではないかと思う。こうなれば J を愛する JAPLA 人として、望外の喜びではないだろうか。

したがって、ここでご披露する VB コードは J 言語としては本質的なものではないのであるが、JAPLA の諸先生方への筆者の研究成果 (?) をそろそろ発表しなければならず、あえて発表させていただくこととした。

5. 1. 本VBプログラムの構造



(Subroutine 等)

jServer : Project Name

frmjServer : J 言語を処理する画面

mduljServer : 画面に依存しない
コード群

5. 2. 画面設計とコード

a. frmjServer



(オブジェクト名)	frmjServer
Appearance	1 - 3D
AutoRedraw	False
BackColor	<input type="checkbox"/> &H8000000F&
BorderStyle	2 - 可変
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - 実線 (中心線)
DrawWidth	1
Enabled	True
FillColor	<input checked="" type="checkbox"/> &H00000000&
FillStyle	1 - 透明
Font	Terminal
FontTransparent	True
ForeColor	<input checked="" type="checkbox"/> &H80000012&
HasDC	True
Height	3540
HelpContextID	0
Icon	(アイコン)
KeyPreview	False
Left	1035
LinkMode	0 - なし

Link Topic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(なし)
MousePointer	0 - 既定値
Moveable	True
NegotiateMenus	True
OLEDropMode	0 - なし
Palette	(なし)
PaletteMode	1 - Z オーダー
Picture	(なし)
RightToLeft	False
ScaleHeight	2730
ScaleLeft	0
ScaleMode	1 - Twip
ScaleTop	0
ScaleWidth	6015
ShowInTaskbar	True
StartPosition	0 - 手動
Tag	
Top	1590
Visible	True
WhatsThisButton	False
WhatsThisHelp	False
Width	6135
WindowState	0 - 標準

プロパティの内容[↑]

```
Private Sub Form_Load()  
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)  
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)  
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 9000)  
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 5000)  
    ,  
    If WindowState = 0 Then  
        Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2  
    End If  
    ,  
    SetDLL  
    Call mnuClearJ_Click  
End Sub  
  
Private Sub Form_Resize()  
    If Me.Width > 6105 Then  
        DoText.Width = Me.Width - (6135 - 5175)  
        GetText.Width = Me.Width - (6135 - 5175)  
        Text1.Width = Me.Width - (6135 - 5175)  
    End If  
    ,  
    If Me.Height > 3510 Then  
        Text1.Height = Me.Height - (3510 - 1695)  
    End If  
    ,  
End Sub  
  
Private Sub DoText_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then  
        KeyAscii = 0  
        ec = Js.Do(DoText)  
        DoText.SelStart = 0  
        DoText.SelLength = Len(DoText.Text)  
        If ec Then  
            Select Case ec  
                Case Is = 16  
                    Text1 = "Value Error (Code=" & ec & ")"  
                Case Is = 3  
                    Text1 = "Domain Error (Code=" & ec & ")"  
                Case Else  
                    Js.ErrorTextM ec, v  
                    Text1 = "Miscellaneous Error (Code=" & ec & ")"  
            End Select  
        ,  
        Else  
            Text2.Text = DoText.Text  
            Text1.Text = Text1.Text & "Do:" & Text2.Text & Chr(13) & Chr(10)  
            Text1.SelStart = Len(Text1.Text)  
        ,  
        End If  
        ,  
        DoText.SelStart = 0  
        DoText.SelLength = Len(DoText.Text)  
        DoText.SetFocus  
    ,  
End Sub
```

```

        End If
    End Sub

    Private Sub SetDLL()
        Set Js = New JDLLServer
        frmJServer.Caption = "J言語 JDLL Server For j601"
    End Sub

    Private Sub GetText_KeyPress(KeyAscii As Integer)
        Dim X As Variant
        Dim v As Variant
        Dim tText As Variant
        If KeyAscii = 13 Then
            KeyAscii = 0
            ec = Js.Do("JGet=" & GetText)
            If ec Then
                Select Case ec
                    Case Is = 16
                        Text1 = "Value Error (Code=16)"
                    Case Else
                        Text1 = "Miscellaneous Error (Code=" & ec & ")"
                End Select
            Else
                ec = Js.GetB("JGet", X)
                If ec Then
                    Select Case ec
                        Case Is = 16
                            Text1 = "Value Error (Code=16)"
                        Case Else
                            Text1 = "Value Error (Code=" & ec & ")"
                    End Select
                Else
                    Text2.Text = ""
                    jDisplay X, Text2
                    Text1.Text = Text1.Text & "Get:" & GetText.Text & Chr(13) & Chr(10) &
                    Text2.Text & Chr(13) & Chr(10)
                    Text1.SelStart = Len(Text1.Text)
                End If

                GetText.SelStart = 0
                GetText.SelLength = Len(GetText.Text)
                GetText.SetFocus
            End If
        End If
    End Sub

    Private Sub mnuClearJ_Click()
        On Error Resume Next
        DoText.Text = ""
        GetText.Text = ""
        Text1.Text = ""
        Js.Clear
        DoText.SelStart = 0
        DoText.SelLength = Len(DoText.Text)
        DoText.SetFocus
    End Sub

```

b. mduljServer

```

Public Js As Object
Public Sub jDisplay(y As Variant, t As Object)
    Dim rank As Long
    rank = getRank(y)
    t = ""
    ,

    If VarType(y) = vbString Then
        t = y
        Exit Sub
    End If
    ,

    If rank = 0 Then
        t = y
    ElseIf rank = 1 Then
        For i = 0 To UBound(y, 1)
            t = t & " " & y(i)
        Next i
    ElseIf rank = 2 Then
        For i = 0 To UBound(y, 1)
            For j = 0 To UBound(y, 2)
                t = t & " " & y(i, j)
            Next j
            t = t & vbCrLf
        Next i
    ElseIf rank = 3 Then
        ,
        For i = 0 To UBound(y, 1)
            For j = 0 To UBound(y, 2)
                For k = 0 To UBound(y, 3)
                    t = t & " " & y(i, j, k)
                Next k
                t = t & vbCrLf
            Next j
            t = t & vbCrLf
        Next i
        ,

    End If
End Sub

```

```

Public Function jDo(txt As String) As Integer
    'execute sentence, return error code
    Dim v As Variant
    jDo = Js.Do(txt)
    If jDo Then
        Js.ErrorText jDo, v
        MsgBox v
    End If
End Function

```

```

Public Function jGet(txt As String) As Variant
    ec = Js.Get(txt, X)
    If ec Then
        Js.ErrorText ec, X
        MsgBox X
    End If
    jGet = X
End Function

```

```

Public Function getRank(v As Variant) As Long
    On Error GoTo Err
    For getRank = 0 To 30
        X = UBound(v, getRank + 1)
    Next getRank
Err:
End Function

```

本プログラムの実行ファイルも添付した。実行ファイルは [jServer.exe]であるが、JDDLserverのランタイムがインストールされていない場合、動作しない可能性があるため、正式なインストーラ[jServer.msi]も添付した。

次回からは、J-DLLserverをインプリメントしたJ言語による統計解析パッケージおよび、ユーザーが解析手法を自作可能なアプリケーションの作成過程を、VBで展開してみたいと思っている。関係諸氏のご意見を頂戴したい。

以上