

Excel_VBA でいろいろな OLE を実験する

一日付時間、電卓、メモ帳、Word から J まで

西川 利男

0. はじめに

J を含めたいろいろなアプリケーションとの間の OLE について最近、例会の話題を賑わしている。[1,2,3] 先月、鳥邊錬太郎氏は VBA の元になる Visual Basic 6.0 の上で J の OLE について、フォームの作成、コードの記述などをていねいに説明され、筆者にとっては極めて有益であった。しかし、残念なことに筆者の手元には非常に古い Visual Basic 2.0 しかないのでトレースすることもままならない。それではと、いっそのこと Excel_VBA の上で違った側面から検討して見るのもまたよし、と考えた。

例題としては、簡単な日付時間表示を手始めとして、電卓、メモ帳から Word、そして J といういろいろな OLE を実験する VBA プログラミングを行ってみた。

Excel_VBA に限ったわけではないが、Windows プログラムは次の 2 つから成っている。

- ・フォーム 処理の入り口であり、また結果表示の出口となるインターフェースとして重要であり、フォームエディタにより作成される。
- ・コード 処理動作の記述であり、狭い意味でのプログラムである。

コードの作成はいうなれば、従来から行ってきた Basic などのプログラミングであるが、フォームの必要性およびその作成は Windows プログラミングに独特のもので、筆者のように古くからのプログラマにとってはようやくその意義を解しやっとなじんで来たものである。その意味では筆者世代の者にとっては目新しいが、若い世代のプログラマではごく当たり前の事柄かもしれない。この点ご容赦いただきたい。

1. VBA の構造と開発の手順

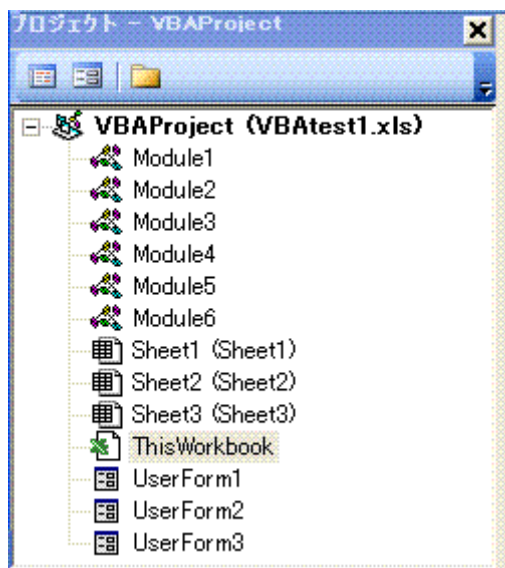
VBA (Visual Basic Application) は Visual Basic と言語文法においては全く同じであるが、Microsoft Word, Microsoft Excel のマクロを記述するためのものであり、システムとしてはいろいろ違っている。さらに VBA 自身も Excel のものと Word のものとは文法や考え方は共通であるが、機能仕様は異なっている。以下ここでは Excel の VBA に絞ってみていこう。

[1] 西川利男「J の Excel_OLE を解析する」JAPL 研究会資料(2007/2/24)

[2] 鳥邊錬太郎「VB-J_DLL Server を使う」同誌(2007/2/24)

[3] 西川利男「J と Word,Excel,HTML－全員集合の協調システム」同誌(2007/1/27)

Excel と Excel_VBA との関係を見るには Excel システム上で Alt-F11 で現れる VBE (Visual Basic Editor) で Project を見ればよく、次のように示される。



全体は VBAProject となっており、例えば VBAtest1.xls のようなファイルである。そして通常の Excel 画面のシートは Sheet1, Sheet2, Sheet3 のようになっている。

いわゆるマクロは Module1, Module2,... であり、ここに VBA プログラムが書かれる。ここにプログラムを書き込むには、VBE のツールバー[挿入]で[標準モジュール]をクリックして現れる Sub ... End Sub のプロシージャとしてプログラミングする。

さらにフォームを必要とするときは[挿入]で[ユーザフォーム]をクリックするとフォームエディタ画面が現れ、同時に現れるコントロール部品の表示から、ボタン、テキストボックスなどをマウスでドラッグして必要な位置、大きさで貼り付ける。また、コントロールのキャプション (名前)、文字の大きさなどいろいろな仕様はプロパティ表に値をセットすることで行う。

今度はボタンなどコントロールによる処理の実行はコントロールをダブルクリックすることでさらに現れるプログラム画面で処理のプログラムを記述する。筆者に言わせればここで初めて従来からなじんだプログラミングを行い、ほっとしたところである。

ところで、マクロを実行するには通常の Excel シート画面上で[ツール]-[マクロの実行]でも行えるが、ここでもフォームを作って、そこで実行する方がっこう良い。それには[表示]-[ツールバー]-[フォーム]により現れる簡易フォームエディタにより、コントロールの作成を行い、これにモジュールのプロシージャ名をリンクさせる。こうすれば、シート上にあるボタンをクリックするとマクロの VBA プロシージャが実行できることになる。

2. 日付・時間の表示

最初は最も簡単な VBA として、日付・時間を出力する関数 Now を呼び出して、メッセージボックスで表示する、というものである。プログラムは次のように一行でよい。

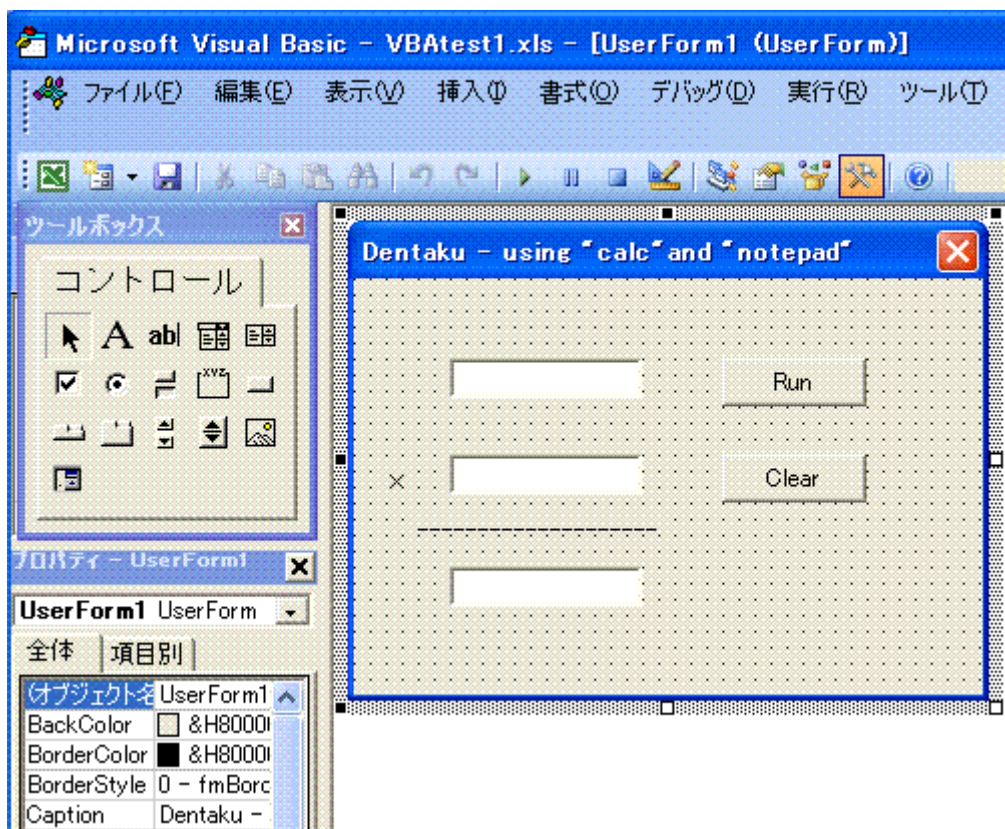
```
Sub ndate()  
    MsgBox "いまの日時は" & Now & "です"  
End Sub
```

3. 電卓とメモ帳を呼び出して処理

今度はフォームを作成し、そのコントロールに VBA のコードを記述する、という通常行う手順にのっとりやってみる。

(1) フォームの作成

VBE エディタのメニューバーの[挿入]で[ユーザフォームの作成]を選んでクリックすれば以下のようにフォームエディタが開かれる。ここでフォーム UserForm1 に、リストボックス、ボタンなどを貼り付ける。掛け算記号、線なども入れて体裁を整える。これらの操作はツールボックスのコントロールのリストからマウスでドラッグ&ドロップで位置、大きさなどを調整して張り付ける。表示(Caption)などはプロパティ表で値を設定する。



(2) コードの記述

コントロールのボタン **Run** を押すと、上の 2 つのリストボックスに入力したデータの値を用いて掛け算を行い、その結果を下のリストボックスに表示するようにする。このとき、掛け算の計算は Windows に常備の電卓プログラムを使い、その結果を一応メモ帳に記録し、最後に Excel に戻るようにした。

上の処理はボタン **Run** を押すと起動されるイベントとして実行されるので、ボタンに付随したコードを記述することになる。このためには、ボタンの上でマウスをダブルクリックすると、コード記述の画面が開きここに VBA のプログラムを書けばよい。

コードは次のようになる。各操作はコメントを見ていただきたい。ここでは Shell により起動し、SendKeys により文字列を送ることで処理を行っている。

Option Explicit

Private Sub CommandButton1_Click()

```
Shell "Calc.exe", 1 'Start calc '電卓プログラムの起動
SendKeys TextBox1.Text, True      'テキストボックス 1 のデータを送る
SendKeys "*", True                '掛け算記号文字を送る
SendKeys TextBox2.Text, True      'テキストボックス 2 のデータを送る
SendKeys "=", True                'イコール記号文字を送る
SendKeys "^C", True               'クリックボードにコピーする(Ctrl-Copy)
SendKeys "%{f4}", True            '電卓プログラムの終了
Shell "Notepad.exe", 1            'メモ帳プログラムを起動
SendKeys "^V", True               'メモ帳に貼り付ける(Ctrl-Paste)
TextBox3.Paste                    'テキストボックス 3 にも貼り付ける
End Sub
```

同様に、ボタン **Clear** に対する VBA コードは以下のようなものである。

Private Sub CommandButton2_Click()

```
TextBox1 = ""
TextBox2 = ""
TextBox3 = ""
End Sub
```

(3) フォームの起動

Excel のマクロとして実行するため、モジュールへの次のプログラムが必要である。

Sub ndentaku()

```
UserForm1.Show
End Sub
```

4. Wordでの実行処理

VBAとしてはたとえ同じでも Word での処理依頼は、それほど簡単ではない。

例題としては文字列を入力して、それを Word に送り、Word 上でその書体、大きさなどを変える、さらに印刷画面のプレビューをするだけのものだが、オブジェクト指向としての前準備、Word_VBA の仕様に合わせるなどの考慮が必要となる。

フォームは省略して、コードのみをあげる。

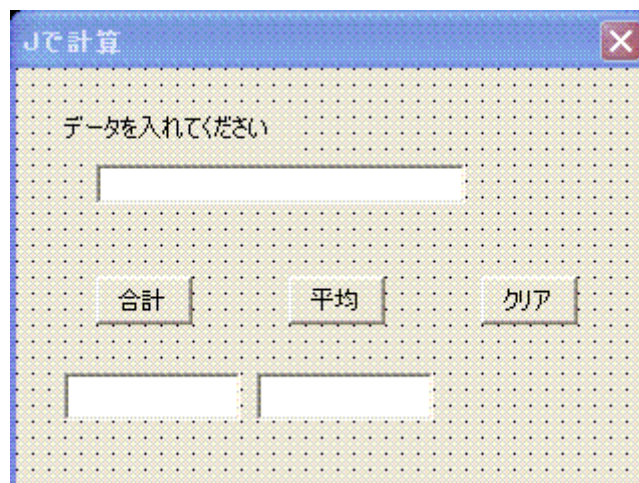
```
Private Sub CommandButton1_Click()
```

```
    Range("A1") = TextBox1.Text           'テキストボックスのデータをセルにコピー  
    Range("A1").Copy                       'セルのデータをクリップボードへコピー  
    Set Obj = CreateObject("Word.Application") 'Word のオブジェクト Obj を作成  
    Set Doc = Obj.Documents.Add           'Documents のオブジェクト Doc を作成  
    With Doc.Paragraphs(1).Range         'Doc.Paragraphs(1)について  
        .Paste                             'クリップボードのデータを貼り付け  
        .Font.Size = 20                    'フォントの大きさを変更  
        .Bold = True                       'ボールドに変更  
        .Italic = True                     'イタリックに変更  
    End With  
    Obj.Visible = True  
    Obj.ActiveDocument.PrintPreview      '印刷プレビューに表示  
    Range("A1").Clear  
    Obj.Quit                               'Word の終了
```

```
End Sub
```

5. Jでの実行処理

次のようなフォームを作った。例題の説明は不要であろう。



3つのボタンのの対する VBA の各コードは次のとおりである。

合計…Sub CommandButton1_Click()

平均…Sub CommandButton2_Click()

クリア…Sub CommandButton3_Click()

コードについてはこれについては何遍も紹介してきたので、説明は省略する。

Option Explicit

Dim N As String, Dim M As Variant

Dim ec, v As Variant

Private Sub CommandButton1_Click()

 N = TextBox1

 jopen

 js.Show 0

 js.Log 1

 jdo "JN=. " & N

 jdo "JM=. +/- JN" 'Only one value can return

 M = jget("JM")

 js.Quit

 TextBox2 = M

End Sub

Private Sub CommandButton2_Click()

 N = TextBox1

 jopen

 js.Show 0

 js.Log 1

 jdo "JN=. " & N

 jdo "JM=. (+/ % #) JN"

 M = jget("JM")

 js.Quit

 TextBox3 = M

End Sub

Private Sub CommandButton3_Click()

 TextBox1 = ""

 TextBox2 = ""

 TextBox3 = ""

End Sub

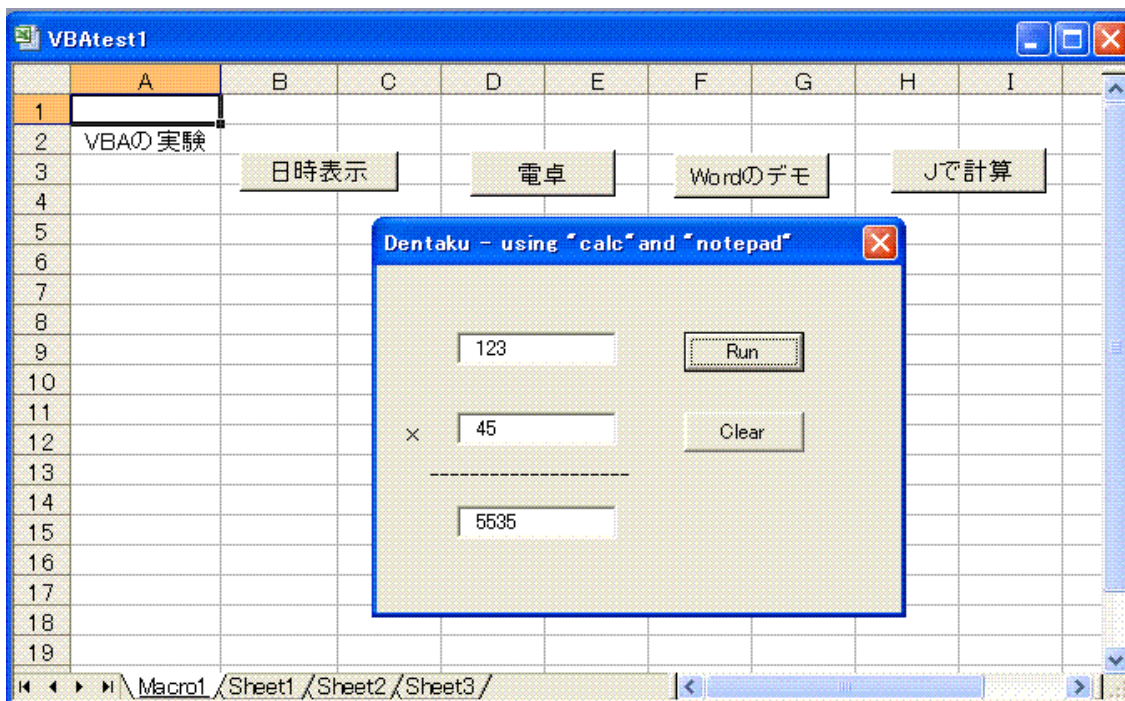
```
Sub jopen()  
Set js = CreateObject("jexeserver")  
End Sub
```

```
Sub jdo(s As String)  
ec = js.Do(s)  
If ec Then MsgBox "Error code: " & Str(ec)  
End Sub
```

```
Function jget(s As String) As Variant  
ec = js.Get(s, v)  
If ec Then MsgBox "Error code: " & Str(ec)  
jget = v  
End Function
```

6. Excel マクロとしての OLE 実行例

実行例として電卓マクロの実際を示す。



7. Excel_VBA と Excel とをこう考えるー筆者の独断的コメント

(1) Excel_VBA の位置づけ

Excel_VBA は一般には Excel の表計算機能を助け、より複雑な高度の処理を行うためのものとされている。しかし筆者の独断的意見として、もっと積極的に Windows 処理の中心技術と位置付けすべきと思う。Visual Basic がその位置にあるようだが、筆者のようなかつての N88BASIC を愛用したユーザにとってはかなりしきいが高い。その一つが入出力インターフェースであるフォームの作成である。この入り口部分は Visual Basic ではなく Excel をそれに当てたらよいと思う。つまり Excel を単なる表計算ソフトではなく Windows プログラミングのインターフェースを含めた汎用ツールと位置づけるのである。

Excel = 配列型の階層構造を持つ汎用オブジェクト・処理ツール

(2) 百人一首の決まり札早取りツール

筆者の娘など、若い世代の Windows ユーザに言わせると、文書は Word ではなく Excel で作るのだという。たしかに章、段落の文章をセルに入れてしまえば、文書の構成編集はずっと楽になる。お正月のお遊びプログラムである。

	A	B	C	D	E	F	G
1	七	六	五	四	三	二	一
2	じ よ の た か き も り え	を わ す ら お も は る ず み	の あ な さ 心 に ね さ し か か お ず や は ら ま ば	も い ま は た え な だ む お	が こ ひ す て だ ふ き	げ や え む ぐ ら し	お ほ わ き ひ が そ い み で し え は の ぬ し
3	も ひ の る を は え こ ぎ そ え お つ も つ	を ひ し と く の な も い あ の ち か の	く ひ と よ に し し も ら が れ な で	い ひ ふ と よ づ し て も な ら が ら な で	お ひ も と ひ し そ れ め ず し こ か そ	あ ひ き と は こ ぎ に み け え り ね	か ひ わ と く こ ま そ も し ら な ら し ね
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

Microsoft Excel

下の句のはじめを入れて下さい

ひとの

OK

キャンセル

(2) 複数の絵画をタテ・ヨコにつなげて大きい作品を創る。

オブジェクトとうたった意味はこの例で示される。Excel では文字列、数学計算値だけではなく画像なども自由に扱える。もう Photoshop などいない。

これは絵を趣味とする妻の要望で作った筆者の秘かな Excel の効用である。

