## JのExcel\_OLE を解析する

## -オブジェクト指向からの理解-

## 西川 利男

## はじめに

先月の例会で筆者は、Excel\_VBAでWordのデータを取り出し、さらにJにより最終的にHTMLファイルを作り出すプログラムについて報告した。

筆者の究極の目標はJのOLE, DDE機能により、Windowsのいろいろなアプリケーションを自由に扱いたいということにある。そのための最初のテーマとしてExcelに対するOLEが最も適当なものと考えた。

幸い、JからExcelのデータを読み書きする処理については、志村正人氏、竹内寿一郎 氏によるていねいな解説がある。[1,2]しかしながら、OLEの操作の詳細については、ブ ラックボックスのようにして使えばよいとして、そのコーディングの意味づけについて は明らかにされてはいない。今回、筆者はこれを元に少し修正し、コーディングの内部に 入り、OLEとしてどのような処理を行っているのかを、理解したいと考えた。

Excel\_VBA はいうまでもなく Visual Basic のプログラミングであるが、データ構造、 処理の理解の基本にはオブジェクト指向の考え方が必要であり、この面からも筆者なり のいささかのコメントをした。

オブジェクト指向とはデータと処理とを別々のものとしてではなく、両方を(これを プロパティ、メソッドと呼ぶが)備えたオブジェクトとして扱うことにある。[3]

オブジェクト 🕂 プロパティ(データ)

名前、構成要素、表示の方法など

- メソッド(処理、関数)

そして、このような機能をクラスという設計図と見て、これからインスタンスという製 品をもらって、これについて処理を行うやり方である、と一口に言えよう。

以上のような理解と実験の過程を、Labシステムにより対話的にJを実行し、体験できるJのExcel\_OLEの解析システム'Analyse\_Excel\_OLE'につき紹介する。

[1] 志村正人「J for Win9x/NT入門、EXCEL とのリンク他」J研究会資料 1998/9/24

[2] 竹内寿一郎「Jの中でエクセル、エクセルの中でJを使う」JAPLA シンポジウム 2000/12/16

[3] 西川利男「Jのオブジェクト指向プログラミングーその1」JAPLA シンポジウム 2005/12/10 JのExcel\_OLEの解析 (以下はLabによる対話出力を印刷したものである)

\_\_\_\_\_

Lab: Analyse J to Excel OLE Author: Toshio Nishikawa

 $\ensuremath{\mathsf{Press}}\xspace$  (Ctrl+A) to advance.

-- (1 of 21) introduction -----JのExcel\_OLEとは

Jから Excel へのやり取りは次のような Client&Server 方式で行う。
 Client: J

Server: Excel

一般には Client&Server 方式とはあらかじめ Server を起動し、受けられる状態にした上で、 Client から Server に処理を依頼する。

しかし、JではClientに特別な機能を持たせて、Serverの起動がな

くても Excel が動くようにした。これが J\_Excel Automation である。・

その上で、JからExcel上で動くVBAマクロ・プログラムを送り込み Excel上でのいろいろな処理を可能にした。

• Excel の構造 Excel は次のような構造を持っている。これはオブジェクト指向からみれば、プロパティとメソッドから成るクラス構造と考えられる。

Application Workbooks Worksheets Charts DocumentsProperties Names

・JのExcel\_OLE 操作と xlutil 関数

基本的には Jの OLE は以下のように

wd 'ole\*\*\*'

oleで始まる命令の文字列をwd(Windows Driver)に渡すことでなされる。

```
しかしながら実際にはいろいろな操作を行うには次のJプログラム
```

examples¥ole¥excel¥xlutil.ijs

で定義されている xl で始まる各種の関数を用いて行う。

詳細はやがてあきらかになるが、これらの命令は常に parent の xlauto に対して行われるので psel(parent select) xlauto が先立つことになる。

xlutilのいろいろな定義はこのカヴァーリング(覆いかぶせ)を行うものである。 たとえば xlget

xlget ARG は wd 'psel xlauto;oleget xl',ARG

のように、内部では oleget の処理を行う。 なお、以下の 3 つの命令が重要である。 xlget 値、オブジェクトなどを得る xlset プロパティに値をセットする xlcmd メソッドを実行する

```
)
```

-- (2 of 21) prepare ------まず準備として excel utitily を読み込む。 )

require 'examples¥ole¥excel¥xlutil.js'

-- (3 of 21) prepare (continued) -------最初に Excel OLE Automation のオブジェクトの作成を行う。

このコーディングの意味は次の通りである。 pc(parent create)により、parent windowとして xlauto を定義し、 cc(child create)により、child controlとして xl を定義する。

オブジェクト指向で考えれば、クラス Excel. Application のインスタンス base が デフォルトとして作られる。

```
その名前をチェックしておく。
)
wd 'pc xlauto'
wd 'cc xl oleautomation:excel.application'
```

```
xlget 'base name'
Microsoft Excel
-- (4 of 21) prepare (continued) -----
この状態をExcel 画面として見てみる。
Alt-F11により VBA-Editor をチェックせよ。まだ何もない。
Excel から戻るときは、最小化中断とすること。
なお、このコードは次の意味である。
オブジェクト base のプロパティ visible に値1をセットする。
これによって Excel のすべての画面が表示可能になる。
ふつうは
   xlshow ''
として行う。
)
  wd 'psel xlauto; oleset xl base visible 1'
-- (5 of 21) prepare (continued) ------
クラス workbooks のインスタンス wb を作成する。
)
  xlget 'base workbooks'
  xlid 'wb'
-- (6 of 21) prepare (continued) -----
ここで、JMACROS. xls なるマクロだけの Excel をオープンする。
つまり、Jのシステムにある Excel に VBA のマクロ・プログラムを送りこむ。
)
  xlcmd 'wb open ', JMACROS
-- (7 of 21) prepare (continued) ------
この状態のExcel 画面を見てみる。
```

Alt-F11により VBA-Editor をチェックせよ。

😤 Microsoft Visual Basic - JMACROS.XLS - [Module1 (3-6)]													
: 🚜 ファイル(E) 編集(E) 表示(V) 打	挿入①	書式(Q) デバッグ(D) 実行(E	) ツール(D) アドイン( <u>A</u> )	ウィンドウ(W) ヘルプ	(H) _ & ×								
図 🛅 + 🖳   糸 釉 礁 船   ヴ べ   ▶ 🗉 🖬 🔟 🧏 🧏 😭 😴 🎌   ⑧   1 行.1 桁													
フロジェクト - imacros.xds X	G	eneral)	<ul> <li>(Declarations)</li> </ul>										
Imacros.xls     (JMACROS.xLS)       Imacros.xls		<ul> <li>J OLE client macros</li> <li>used by script: exampl see this script for mo</li> <li>Macros:         <ul> <li>jread, jreadr</li> <li>jwrite, jwriter</li> <li>jsetchart</li> <li>jquit</li> <li>quit</li> <li>texamples:</li> <li>read from bookl, sheet</li> </ul> </li> </ul>	es¥ole¥excel¥xlutil. re information ad cell, read range ite cell, write rang t chart data it Excel (no save) 1, the 2 by 3 range	js e starting at 1 1:									
く Module1 × 2日パディーModule1 × Module1 Module ・ 全体 項目別1 ダブジェクト名)Module1		<ul> <li>xIreadr 'bookl sheet</li> <li>write to cell 2, 2 val</li> <li>123 xIwrite 'bookl s</li> <li>write 2 by 3 range sta</li> <li>data xIwriter 'bookl</li> <li>set chart source:</li> <li>xIsetchart 'chartl s</li> <li>Notes:</li> <li>Test a macro in Excel</li> <li>wd'ger' reporting macr</li> <li>All macros in a module</li> <li>before any macro will</li> <li>for a J call only reporting the macro in Excel reporting</li> <li>the macro should be d</li> <li>jread book sheet row c</li> <li>Function jread(b, s, r A</li> <li>jread = Workbooks(b).Wor</li> </ul>	1 1 1 2 3' ue 123: heet1 2 2' rting at 1 1: sheet1 1 1 2 3' heet1!A3:C8' before trying it fro o not found is a com must be successfull be found. A compilat rts macro not found. orts the compilation efined as Function a ell s Integer, c As Inte ksheets(s).Cells(r,	m J. mon problem. y compiled ion failure Running problem. <u>s String</u> c).Yalue									

jmacro.xls がロードされ、module1のVBA プログラムを確認せよ。

Excel から戻るときは、最小中断とすること。

```
ここまでの操作はふつうは

xlopen ''

としてまとめて行われる。

)

xlshow ''

-- (8 of 21) prepare (continued) ------

空の Excel に追加により、新規にブック wa を作る。

オブジェクト wa の名前をチェックする。

)

xlcmd 'wb add'
```

```
xlid 'wa'
```

```
xlget 'wa name'
Book1
-- (9 of 21) prepare (continued) ------
そして Excel 画面を見てみる。
)
  xlshow ''
-- (10 of 21) prepare (continued) -----
waをファイル"TEST. xls"に書き出す。
)
  xlcmd 'wa saveas "TEST.xls"'
-1
-- (11 of 21) prepare (continued) -----
この状態のExcel 画面を見てみる。
)
  xlshow ''
-- (12 of 21) prepare (continued) -----
waのプロパティ worksheets を取り出し、オブジェクト ws とする。
)
  xlget 'wa worksheets'
  xlid 'ws'
-- (13 of 21) prepare (continued) -----
wsのプロパティ item, sheet1 をとり、オブジェクト sh1 とする。
この名前をチェックする。
)
  xlget 'ws item sheet1'
  xlid 'shl'
  xlget 'sh1 name'
Sheet1
```

```
-- (14 of 21) prepare (continued) ------
オブジェクト sh1 の名前を Sheet1 から TRY に替える。
そして、オブジェクト sh1 をアクティブにする。
また、Excel 画面でチェックする。
)
  xlset 'sh1 name TRY'
  xlget 'sh1 name'
TRY
  xlcmd 'sh1 activate'
-1
  xlshow ''
-- (15 of 21) prepare (continued) -----
ここで、現在アクティブのブック、シートの名前を確認する。
)
  xlwbws ''
+----+
TEST. x1s TRY
+----+
-- (16 of 21) calc -----
さてここで、JとExcelとでやり取りするデータをつくる。
)
  Dat0 =: i. 3 4
  Dat0
0 1 2 3
4 5 6 7
8 9 10 11
-- (17 of 21) calc (continued) ------
この値を Excel のデータとして、ファイル
 TEST.xls
のシート
```

xlshow ''

🗷 Microsoft Excel - TEST 📃 🔲 🖂												
■ ファイル(E) 編集(E) 表示(V) 挿入(P) 書式(Q) ツール(T) データ(D) ウィンドウ(W) ヘルブ(H) = ■ ×												
<mark>i</mark> MS I	Pゴシック	<b>-</b> 11 -	BIU		🔤 i 📆 %	, :.0 1	E   🛄 + 🔕	• <u>A</u> • 🚆				
A1 🚽 🔊 🖉 🖉												
	A	В	С	D	E	F	G	н 🗖				
1	0	1	2	3								
2	4	5	6	7								
3	8	9	10	11								
4					-							
5												
6												
7												
8												
9												
10												
11		01 10 /01						<u>×</u>				
I												
ועדב	ς			合計=6	6							

-- (19 of 21) calc (continued) -----

次にこの Excel データの

セルA2 つまり行2,列1 の位置から

2行、3列 の値を読み出し、Jの値にする。

```
xlreadrは範囲(range)を指定しての読み出しである。
)
  Dat1 =: xlreadr 'TEST.xls TRY 2 1 2 3'
  Dat1
+-+-+--+
456
+-+-+--+
8910
+-+-+--+
-- (20 of 21) calc (continued) ------
最後にExcel 画面をチェックする。
今度はExcelを終了して、閉じる。
)
  xlshow ''
-- (21 of 21) fin -----
最後に Excel_OLE を終了する。
)
  xlquit ''
End of lab
```