

# Getexcel.ijs の最近の情勢について

A Report of the Recent Use of Getexcel.ijs

(株) 竹内八ガネ商行  
竹内寿一郎

## 1 . はじめに

志村氏が 1999 年 12 月のシンポジウムで、J の OLE を立ち上げて J の中でエクセルを使う方法を論じた [1]、[2] のに刺激を受け、2000 年 12 月のシンポジウムで私が他人の禪で相撲を取るように発表したのが「J の中でエクセルをエクセルの中で J を使う」であった [3]。当時それ程検討せずに志村氏の論文をもとに見よう見まねで書いたものであり、このとき陽に Getexcel.ijs が紹介され、その後皆さんに便利に使われてきた。ところがどういう訳かこの関数は私が作ったように JAPLA 研究会でいわれてきたがいつも何かすぐたい思いで一杯だった。Getexcel.ijs を歴史的にみると、志村氏がまず JAPLA 研究会で紹介し、私が少し手を加えて、J からエクセル上の数値を簡単に読み書きできるような関数 datain、dataout という名前で広めたものだった。

これらの関数は残念ながら数値しか扱えなかったのが最大の欠点で (私の手の加え方が不十分だったため)、さらに志村氏が 2005 年 10 月の例会でお仕事の関係で大量のデータ (csv データ) をエクセルから J に読み込むには readcsv. ijs で 10 万 ~ 20 万件くらい簡単に読めることを話された。と、またここで刺激を受け Getexcel.ijs を修正して、charin という関数をつくりエクセルシートから数値に限らずどんな文字でも読み込めるようにし、またエクセルに限らず、テキストファイルを読み書きするための J の関数 readtable、writetable を Getexcel.ijs に組み込んだのが 2005 年 11 月の「エクセルと文字列をデータ交換する関数 Getexcel.ijs」であった [4]。

そこで今回、志村氏の薦めもあって J でのエクセル関数について少し詳しく調べてみることにした。

## 2 . Getexcel.ijs の現況

実は、J501 以降の Getexcel.ijs はいろいろ手を加えなければならない状況にあった。それは /J501/system/examples/ole/excel/ の中に jmacros.xls が欠けていたため Getexcel.ijs が動かなかったのである。そこでこれまでは応急処置として J401 などから jmacros.xls をコピーしてそのディレクトリーに入れて動かしてきたのである。ところがそれをよくよく調べてみたら、Getexcel.ijs にある xlutil. ijs という J の関数群が働くには jmacros.xls のマクロとして定義される Visual Basic がなければ動かない状況にあったのである。

NB. get EXCEL use OLE (xlutil. ijs)

getexcel=: 3 : 0

NB. J Client use ole to EXCEL Sever

NB. Filename of Book is TEST and name of Sheet is TRY

NB. Usage: getexcel ''

```

load'c:\j503\system\main\files.ijs'
load'c:\j503\system\main\stdlib.ijs'
load'c:\j503\system\examples\ole\excel\xlutil.ijs' NB. adjust your directry
wd 'reset'
xlopen ''
xlshow ''
xlcmd 'wb add'
xlcmd 'temp saveas TEST'
xlget 'temp worksheets'
xlid 'ws'
xlget 'ws item sheet1'
xlid 'sh1'
xlset 'sh1 name TRY'
xlcmd 'sh1 activate'
)}

```

これは j503 で使っていた Getexcel.ijs である。上の関数の  
load'c:\j503\system\examples\ole\excel\xlutil.ijs' NB. adjust your directry  
は J の置かれた環境によって変えなければならないのであるが、jmacros.xls は J の初期状態  
ではこのディレクトリーに存在せず、J403 のシステムからコピーしたものである。  
次に Getexcel.ijs の中で読み込んでいる、xlutil.ijs の一部を示す。

NB. utilities:

```

NB. xljmacro      run macro from jmacros.xls
NB. xlwbws       return active workbook, worksheet
NB. xlbox        box argument

```

NB.

NB. utilities from standard J library:

```

NB. clipfmt      format data for clipboard
NB. clipunfmt    unformat data read from clipboard

```

```
JMACROS=. jpath '~system\examples\ole\excel\jmacros.xls'
```

NB. =====

```

xlget=: wd@('psel xlauto;oleget xl '&,)
xlset=: wd@('psel xlauto;oleset xl '&,)
xlcmd=: wd@('psel xlauto;olemethod xl '&,)
xlid=: wd@('psel xlauto;oleid xl '&,)

```

NB. =====

```

xlopen=: 3 : 0
if. -. 1 e. ('xlauto',LF) E. wd 'qp' do.
  wd 'pc xlauto'
  wd 'cc xl oleautomation:excel.application'
  wd 'oleget xl base workbooks;oleid xl wb'
  wd 'olemethod xl wb open ',JMACROS
end.
empty''
)

```

この中で xlopen という関数のところで JMACROS という名で jmacros.xls に書かれた Visual Basic を読み込むようになっている。ここで JMACROS で定義される jpath とは J が置かれたフォルダ名で、これまではルートに置かれていたので c:\J\*\*\*a とか c:\J\*\*\* という名が jpath であった。

ところが、J601 以降、J は必ずしもルートにある必要は無く、しかも個人用とユーザー全員用では J が登録される場所が異なり、デフォルトでは、個人用では Documents and settings\ユーザー名\j602-user、全ユーザー用では Program Files\j602 になっているので、それを無理に変えるとあとが厄介になる。

これまでは J のルートフォルダに temp があるので使い易かったが、J を個人用にインストールするとして、例えばシステムは Program Files に無理に入れようとする、システムフォルダは Program Files\ となるが、temp フォルダ、config フォルダ、lab フォルダは Documents and settings というように分かれてしまっていて始末に悪い。どうやらデフォルト通りにインストールした方が無難である。しかも出来るだけ個人用は後が使いにくいので、そこにインストールしないようにした方が良いでしょう。

### 3 . J OLE Client to Excel の解説 ( 1 )

J からエクセルを使うチュートリアルは \* \* \* \* \* にあり、Jsoftware.com からダウンロードできるが J の Help から手に入れる方が手っ取り早い。

J では VB のような階層をもった名前がないので次のようにする。

エクセルでは次のような階層の width を定義するが

Activesheet.ChartObjects.Item(1).Chart.PlotArea.Width のことを J では階層という概念がないので 2 つの名前で表現する。abc は何でもよい。

```
abc width
```

エクセルで画面に出すとき

```
Application.Visible
```

とするが、J では

```
base visible とする。
```

ここで base とは J では階層のトップまたはエクセルにおけるアプリケーションと思ってよい。

Worksheets("Sheet1").Activate で新しいワークシートをつくるが、J ではテンポラリに temp という名のシートができ、実際には temp は Sheet1 という名でシートが出来る。

```
temp activate
```

実際に、任意の時点ではテンポラリーではあるが、そのときのテンポラリーシートに名前をつけておくと、以後それを名指しでつかうことができ、

```
sh1 activate
```

として使うことができる。

簡単な使用例：

```
wd 'pc xlauto'
```

OLEを立ち上げ、エクセルプログラムをメモリに取り込む。

```
wd 'cc xl oleautomation:excel.application'
```

以後JのOLEコマンドxlで表す宣言をする。このときxlではなく何の文字を使ってもよいが、ここでのJで使われるコマンドにこれが使われているので、xlを使うようにした方がよい。

VBの

```
Application.Visible = 1
```

はJではアプリケーションオブジェクトの名はbaseなのでそれを見えるようにvisibleに1をセットして

```
wd 'oleset xl base visible 1'
```

というわけでエクセルが見えるようになる。しかしまだブックが開かれたわけではない。ここでワークブックオブジェクトを加えるには

```
wd 'oleget xl base workbooks'
```

を始めに実行する。そのあと

```
wd 'olemethod xl temp add'
```

とするとBook1が作られる。更にいくつかのブックを追加することが出来るが wd 'olemethod xl temp を繰り返してはエラーとなる。tempはあくまでも名前でなくテンポラリーだからである。そこで

```
wd 'oleget xl base workbooks'
```

```
wd 'oleid xl wb'
```

```
wd 'olemethod xl wb add'
```

```
wd 'olemethod xl wb add'
```

```
wd 'olemethod xl wb add'
```

とすればいくつでもブックを追加することができる。ただしブック名はBook1、Book2、Book3である。

エクセルを閉じるには

```
wd 'olemethod xl base quit'
```

を使う。ただし、ブックを保存しないので注意を要する。従って本当に閉じる前に保存するかどうかJに戻ってプロンプトを用意すべきである。

最後にJウィンドウドライバーをリセットするには

```
wd'reset'
```

として終了する。

### 3 . J OLE Client to Excel の解説 ( 2 ) xlutil.ijs

実際に J の ijs 画面でエクセルを使うコマンドは Utilities.ijs に書かれていて、  
load 'system\examples\ole\excel\xlutil.ijs'  
names ''

として調べてみよう。ここでは xl で始まるそのいくつかを紹介する。

xlopen

create Excel OLE automation object

xlshow

show/hide Excel OLE automation object

xlexit

exit Excel OLE automation object (saves)

xlget

cover for oleget - get object

xlset

cover for oleset - set object parameter

xlcmd

cover for olemethod - invoke method

xlid

cover for oleid - assign id to current position

xlread

read cell

xlreadr

read range

xlwrite

write cell

xlwriter

write range

```
xlsetchart
  set chart range
```

The verb xlopen opens up Excel. It:

```
creates the parent window xlauto
```

```
creates the Excel OLE automation control xl
```

```
names wb as the Workbooks object
```

```
loads the macro file examples\ole\excel\jmacros.xls (which is hidden)
```

Try:

```
xlopen''
```

Note that Excel is not shown, indeed you may want to use Excel without it ever being v

```
xlshow''
```

Verb xlcmd runs an OLE method. Since wb has been named in xlopen, it can be used direc

```
xlcmd 'wb add'
```

Take a look at the workbook name:

```
  xlget 'temp name'
Book1
```

Try changing the workbook name:

```
  xlset 'temp name Mybook'
|domain error
| xlset'temp name Mybook'
```

```
  wd 'qer'
ole - Workbook does not have writeable Name property : 12
```

What is happening is that in Excel, you can only change the name of a workbook by sav

```
xlcmd 'temp saveas Mybook'
```

This may return -1, which really is the result from Excel!

(If you already have saved Mybook, Excel will prompt you to overwrite it.)

Accessing the Worksheet

To access the worksheet, we first have to get the Worksheets object, which belongs to

```
xlget 'temp worksheets'
xlid 'ws'
```

We can try adding new worksheets:

```
xlcmd 'ws add'
xlcmd 'ws add'
xlcmd 'ws add'
```

Next we access the first sheet using the Item method, and assign the name sh1:

```
xlget 'ws item sheet1'
xlid 'sh1'
xlget 'sh1 name'
Sheet1
```

Be careful to distinguish sh1 which is the name used by J for a position in the Excel

```
xlset 'sh1 name Mysheet'
```

If this worksheet is hidden behind another (which will be the case if you followed the

```
xlcmd 'sh1 activate'
```

Now lets try writing to a specific cell. In Excel you can use cell references of the

```
xlget 'sh1 cells 2 3'
```

Then set the value of temp as required. The new value should appear in the spreadsheet

```

xlset 'temp value 123'
xlget 'temp value'
123

```

#### 4 . Getexcel in J602

これまでくどくどと述べてきたことをまとめて以下のような新しい Getexcel.ijs を紹介する。これまでに較べて J が何処にインストールされるか分からないので、新たに jpath が所々に導入されたのが 1 つの新しい点で、もう一つは 1 つのセルから読み書きするときの関数 getcell、putcell を追加したことにある。

```

NB. =====
NB. get EXCEL use OLE (xlutil.ijs)
getexcel=: 3 : 0
NB. J Client use ole to EXCEL Sever
NB. Filename of Book is TEST and name of Sheet is TRY
NB. Usage: getexcel ''
O!:0<jpath '~system\main\files.ijs'
O!:0<jpath '~system\main\stdlib.ijs'
load jpath '~system\examples\ole\excel\xlutil.ijs' NB. adjust your directry
wd 'reset'
xlopen ''
xlshow ''
xlcmd 'wb add'
xlcmd 'temp saveas TEST'
xlget 'temp worksheets'
xlid 'ws'
xlget 'ws item sheet1'
xlid 'sh1'
xlset 'sh1 name TRY'
xlcmd 'sh1 activate'
)
    getcell=:3 : 0
xlget 'sh1 cells ',":y
xlget 'temp value'
)
    putcell=:4 : 0
xlget 'sh1 cells ',":y
xlset 'temp value ',":x
)
    datain=:3 : 0
NB. a=.datain 1 1 5 5 (1 1) denotes a location; (5 5) denotes a size

```



```
(_2{.y})$,0".>xlreadr 'TEST.xls TRY ',":y
)
```

```
dataout=:4 : 0
```

NB. c dataout 6 6 (6 6) denotes a location of Excell Files

```
x xlwriter 'TEST.xls TRY ',":y
)
```

```
charin=:3 : 0
```

NB. If you want to read characters, Use the "charin" function

NB. a=.charin 1 1 5 5 (1 1) denotes a location; (5 5) denotes a size

NB. Results are expressed by a Boxed Form

```
xlreadr 'TEST.xls TRY ',":y
)
```

```
readtable=:[:>0:".&.>[:cutopen([: 1!:1 ])-.(13{a.)"_
writetable=:4 : '(,("x), "1 CRLF)1!:2 y'
```

NB. Examples; makes a matrix file after reading

NB. from Excell File "Sheet TRY" in "TEST.xls"

NB. a=.datain 3 2 32 13

NB. a writetable<'c:\j503\temp\test.dat'

#### 【参考文献】

- 【1】志村正人 (1999): Jでデータ処理 JAPLA シンポジウム 1999 於: 統計数理研究所  
1999.12.11 シンポジウム資料
- 【2】志村正人 (1999): Data processing using J part II JAPLA シンポジウム 1999 於:  
統計数理研究所 1999.12.11 シンポジウム資料
- 【3】竹内寿一郎 (2000): Jの中でエクセルをエクセルの中でJを使う JAPLA シンポジ  
ウム 2000 於: 統計数理研究所 2000.12.16 シンポジウム・チュートリアル資料
- 【4】竹内寿一郎 (2005): エクセルと文字列をデータ交換する関数 Getexcel.ijs JAPLA 研  
究会 2005.11.26 資料
- 【5】J Software(2008) : J OLE Client to Excel  
\j602\help\usr\tutorial\_ole\_client\_excel.htm