

# How To Read and Write in Excel Format with G7

September 10, 2006

This document describes the capabilities of G7 to read and modify files that are in the Microsoft Excel format. The tools allow data to be read from Excel files and written to the G7 workspace data bank or to a VAM file. Also, text may be read and then recorded using the G7 “catch” command for later processing. Data can be written from G7 databanks to existing Excel files, but new files and worksheets cannot be created by G7.

This document provides details for the G7 scripting language. In addition, sample code with comments is provided below. A zipped file is available that contains G7, this documentation, and demonstration files similar to those below.

Use of these commands requires that G7, Version 7.3685 or later, be installed on your computer. In addition, Microsoft Excel 2000 or later is required for use of these tools.

These capabilities are new for G7, and hence more testing and debugging is required. When you run into problems, and if G7 refuses to run any “xl” commands, then open the Windows Task Manager by right-clicking on the Task Bar and selecting “Task Manager” from the menu. Click on the Processes tab and select “Excel” from the list. Click “End Process” and then “OK.” This will shut down the Excel server and should enable G7 to restart Excel and process your script.

Ron Horst

## Example 1

Demo file should be installed in the demo1 directory.

```
vamcr vam.cfg hist
vam hist b
dvam b
```

### # MAKE SAMPLE DATA

```
fdate 1975 2010
f t = 1974 + @cum(t, 1.0, 0.0)
f y = 1975 / (t - 1975)
f y{2010} = -0.0000001 # Set y{2010} to missing value
```

### # WRITE TO EXCEL FILE. NOTE: PRESENTLY, WORKBOOKS AND WORKSHEETS # CANNOT BE CREATED BY G7!

```
xl open xltest.xls # Start or attach to Excel server, and open
# the xltest.xls workbook.
```

```
xl open worksheet 1 # Open worksheet 1.
xl visible # Make worksheet visible
```

```
xl write A 1 "Writing text to file:" # Write a string to position A1.
xl write A 2 "Cell 1,2" # Write a string to position A2.
xl write A 3 "Cell A3" # Write a string to position A3.
xl write B 2 "Cell B2" # Write a string to position B2.
xl write 2 3 "Cell B3" # Write a string to position B3.
```

```
xl open worksheet 2 # Open worksheet 2.
xl write A 1 "Record data columns" # Record label for the date
xl write A 3 "Year" # Record the date
xl write A 4 down t 1976 2010 # Record series name
xl write B 3 "y" # Write series 'y' from the workspace to the
# worksheet, starting in cell B4 and moving
# down the sheet, for years 1976-2010.
```

```
xl write E 1 "Record data in rows"
xl write E 3 "Year"
xl write F 3 right t 1976 2010 # Write series dates from the
# workspace to the worksheet,
# starting in cell F3 and moving
# across the sheet, for years
# 1976-2010.
```

```
xl write E 4 "y"
```

```
xl write F 4 right y 1976 2010
```

```
# Write series 'y' from the workspace to the  
# worksheet, starting in cell F4 and moving  
# across the sheet, for years 1976-2010.
```

```
xl write AN 4 "N/A"
```

```
# Record "Not Available" code for  
# demonstration
```

```
xl close
```

```
# Close the workbook.
```

## ## READ EXCEL FILE

```
xl open xltest.xls
```

```
xl invisible
```

```
# Open the workbook.
```

```
# Make worksheet invisible; this step is  
# redundant, since Excel is invisible by  
# default
```

```
xl replace 0.0
```

```
XL missing "N/A"
```

```
# Set replacement for missing values
```

```
# Set the code for missing values in the  
# Excel file.
```

```
xl open worksheet 2
```

```
xl read A 1 ""
```

```
# Open worksheet 3.
```

```
# Read the string in position A1; string will  
# be printed on screen.
```

```
xl read 2 3 ""
```

```
# Read the string in position B1.
```

```
xl read B 4 down b.y1 1976 2010
```

```
# Read data into vam bank b.
```

```
xl read F 4 right y2 1976 2010
```

```
# Read data into workspace.
```

```
xl exit
```

```
# Close workbook, close connection to  
# Excel server.
```

## # CHECK DATA IN G7 DATA BANKS

```
ic Reading from Vam file
```

```
ty b.y1 1976 2010
```

```
ic Reading from the workspace
```

```
ty y2 1976 2010
```

### Example 2

Demo file should be installed in the demo2 directory.

# The data is read from BEA\section2all\_xls.xls and then is stored in G banks.

[illegible]

### Example 3

Demo file should be installed in the demo3 directory.

ic A series of xl reads that skips over the missing lines in the  
ic Excel worksheet:

```
vamcr gdp.cfg gdp
vam gdp c
dvam c
```

```
xl open c:\test\demo4\C0301e.xls
xl open worksheet 1
```

```
fdate 1990 1990
do{
  xl read %1 27 down c.gdpN%2 1990 1990
  }(3-4 6-7 9-10)(1-6)m
show gdpN
```

```
fdate 1991 1995
do{
  xl read %1 29 down c.gdpN%2 1991 1995
  }(3-4 6-7 9-10)(1-6)m
show gdpN
```

```
fdate 1996 2000
do{
  xl read %1 29 down c.gdpN%2 1996 2000
  }(3-4 6-7 9-10)(1-6)m
show gdpN
```

```
xl exit
```

## Example 4

Demo file should be installed in the matread directory.

```
# matread

vamcr AM.cfg AM
vam AM c
dvam c

xl open C0319e.xls
xl invisible
xl open worksheet 1

# NOTE: To convert empty cells to zero use the following:
xl replace 0.0

show AM y 2000
ic Read a single column, store in matrix.
xl matread c(2) r(14-17, 19-20, 22-24,26-29,31-32, 34-35) c.AM c(1)
r(1-17) 2000
show AM y 2000

ic Read a single row, store in matrix.
xl matread c(2-18) r(14) c.AM c(1-17) r(1) 2000
show AM y 2000

close c
vamcr AM.cfg AM
vam AM c
dvam c

show AM y 2000
ic Read a full matrix from Excel, store in Vam matrix.
xl matread c(2-18) r(14-17, 19-20, 22-24,26-29,31-32, 34-35) c.AM
c(1-17) r(1-17) 2000
show AM y 2000

xl exit
```

## Example 5

Demo file should be installed in the `matwrite` directory.

```
vam AM c
dvam c

show AM y 2000

xl open AM2000.xls
xl visible
xl open worksheet 1

fdate 2000 2000

do{
  do{
    xl write %1 %3 down c.AM%4.%2
    }%1 %2 (4-20)(1-17)m # Excel rows and vam rows, respectively.
  }(B-R)(1-17)m # Excel columns and vam columns, respectively.

xl exit
```

# Syntax

## xl clear missing

This command clears missing value codes. It also resets the replacement value to the default.

Examples:

```
xl clear missing
```

## xl close

This command closes an open workbook. It does not shut down the Excel session that is running in the background.

Examples:

```
xl close
```

## xl exit

This command closes an open workbook and disconnects G7 from the Excel server that is running in the background. (Note: the Excel server may continue running; it can be closed by opening the Task Manager, selecting “Excel,” and terminating the process.)

Examples:

```
xl exit
```

## xl invisible

This command makes invisible the Excel program, provided that Excel is running. Making the display invisible should increase the execution speed of your script.

Examples:

```
xl invisible
```

## xl matread <XL cols><XL rows><matrix><Vam cols><Vam rows><Period>

**XL Columns:** A listing of columns to be read in Excel file. May be given as letters or numbers  
e.g. (1-2, 5, 26-28) or (A-B, E, Z-AB).

**XL Rows:** A listing of rows to be read in Excel file. Must be given as numbers, e.g. (1-2, 5, 8).

**Matrix:** The Vam matrix that will be used to store the data.

**Vam Columns:** A listing of columns to be written in the Vam file. Must be given as numbers  
e.g. (1-2, 5, 8).

**Vam Rows:** A listing of rows to be written in the Vam file. Must be given as numbers,  
e.g. (1-2, 5, 8).

**Period:** The date, where the date is in G7 date format.

This command reads data from an open Excel worksheet and records it as a Vam matrix.



Examples:

```
xl matread (A-D, F) (1-20) Y (1-5) (10-30) 2000
xl matread (1-4, 6) (1-20) Y (1-5) (10-30) 2000
```

`xl missing [ arg1 ]`

This sets a missing value symbol for the Excel spreadsheet. When G7 is reading the Excel file, a “missing value” entry is recorded in the G7 data bank for any Excel cell containing this symbol. The symbol may be a word, number, or a string contained in quotes. Up to 10 missing value codes may be stored, but each must be entered separately. If the command is given without arguments, then previous entries are reported. The following example allows G7 to recognize 0.0, NA, and \_N/A\_ as missing value codes when they are read from an Excel file. See also the `replace` command.

Examples:

```
xl missing 0.0
xl missing NA
xl missing _N/A_
xl missing
```

`xl open`

`xl open workbook <arg1>`

`xl open worksheet <arg1>`

An “open” command with no arguments launches the Excel server.

An “open workbook” command must be provided the name of an existing Excel file.

An “open worksheet” command must be provided the number of a worksheet, where the worksheets are numbered beginning with 1. The instruction “worksheet” may be replaced with “ws,” and the instruction “workbook” may be omitted or replaced with “wb.”

Examples:

```
xl open
xl open workbook c:\test\demo1\xltest.xls
xl open worksheet 1
```

`xl replace <arg1>`

This command sets a replacement value for missing values in the Excel file. If G7 reads a value that matches an entry set in “xl missing,” then the value is replaced by `arg1`. By default, this replacement value is the G7 missing value code (-0.0000001). `Arg1` must be a number. The following example replaces missing value codes read from an Excel file with zeros in the G7 data bank. See also the `missing` command.

Examples:

```
xl replace 0.0
```

`xl save worksheet <name>` (*Not implemented*)

A “save worksheet” command must be provided a name for the open worksheet. “Worksheet” may be abbreviated “ws.”

Examples:

```
xl save ws "Readme"
```

`xl visible`

This command makes visible the Excel program, provided that Excel is running. Making Excel visible may decrease the execution speed of your script.

Examples:

```
xl visible
```

`xl write < column > < row > < value >`

`xl write < column > < row > < direction > < series > [< start > [ end ]]`

Column: May be given as a number or as letters (e.g. 1, A, AF).

Row: A number.

Value: May be any text entry provided in quotes.

Direction: Either d (down) or r (right) if data is to be written in a column or row, respectively, starting with the cell specified with the row and column entry.

Series: A G7 data series. Bank letters are allowed, so the series may be from the workspace bank, a macro bank, or an element of a vector or matrix.

Start: The starting date, where the date is in G7 format. If no start date is provided, the desired start date is assumed to be that set by the current “tdates” setting.

End: The ending date, where the date is in G7 format. If no start date is provided, the desired start date is assumed to be that set by the current “tdates” setting.

This command writes text or data to an open Excel worksheet.

Examples:

```
xl write E 3 "Year"
```

```
xl write AF 1 right a.y
```

```
xl write 1 1 down gdp 1976 2010
```

`xl read < column > < row > < value >`

`xl read < column > < row > < direction > < series > [< start > [ end ]]`

Column: May be given as a number or as letters (e.g. 1, A, AF).

Row: A number.

Value: To read a text entry, provide a set of quotes. The value is displayed on the G7 output window. If “catch” is on, then the value also is stored as text.

Direction: Either d (down) or r (right) if data is written in a column or row, respectively, starting with the cell specified with the row and column entry.

Series: A G7 data series. Bank letters are allowed, so the series may be written to an element of

a vector or matrix in the default Vam bank. If no letter is provided, then the data is recorded in the workspace bank.

Start: The starting date, where the date is in G7 format. If dates are not provided, the desired start date is assumed to be that set by the current “fdates” setting.

End: The ending date, where the date is in G7 format.

This command reads text or data from an open Excel worksheet.

Examples:

```
xl read E 3 " "  
xl read AF 1 right a.y  
xl read 1 1 down gdp 1976 2010
```

`xl vecread < c ( cols ) > < r( rows ) > < direction > < vector > ....  
< v ( index ) > [< start > [ end ]]`

Cols: Columns in Excel file. May be given as a number or as letters (e.g. 1, A, AF).

Rows: Rows in Excel file.

Direction: Either d (down) or r (right) if data is written in a column or row, respectively, starting with the cell specified with the row and column entry.

Vector: A G7 vector. Bank letters are allowed.

Start: The starting date, where the date is in G7 format. If dates are not provided, the desired start date is assumed to be that set by the current “fdates” setting.

End: The ending date, where the date is in G7 format.

This command reads vector data from an open Excel worksheet.

Examples:

```
xl vecread c(c-d f-g) r(35) down c.gdpN v(1-4) 1996 2000
```

## Known Bugs

Unfortunately, G7 sometimes does not handle properly the discovery of an empty cell. If you have trouble reading data from a file, please check your Excel file for empty cells. If such cells exist, try either adjusting your G7 script to avoid the empty cells or replacing the empty cells with a missing value code. If this does not solve the problem, check your script. If the problem persists, please report it.