Calling Ox from Excel

Jurgen A. Doornik

January 10, 2022

Contents

1 The excel2ox Ox package

This package provides XLL executable files that allow Ox code to be called from an Excel worksheet.

1.1 Version

This is version 9.0 of the excel2ox Ox package. (The previous version was 2, but I changed this to follow the Ox version number.)

1.2 XLL

The XLL is a dynamic link library using the `.xll` instead of the `.dll` extension. This can be installed in Excel to provide functions that can be called from a spreadsheet. We use Excel-dna (`github.com/Excel-DNA`) to create the xll, with the Ox wrapper code written in C#.

1.3 Requirements

32-bit Excel for Windows This is no longer supported.
64-bit Excel for Windows

- 64-bit Excel from Office 365, Excel 2019
  To check the precise version use File>Account>About Excel
- Ox 9
- the `OxMetrics9\ox` folder has to be in the search path. Note that this is not set by default, to set it use Advanced system settings>Environment variables>User variables>Path.

To check, type `oxl` in a command prompt window to see which version (if any) of Ox is in the search path.

If Ox cannot be found, or the 64-bit version is used with 32-bit Excel, it is likely that Excel will crash! Or you may get the following, rather confusing message when 32-bit Excel is started:

> The file format and extension of 'excel2*.xll' don't match. The file could be corrupted or unsafe. ...
> Do you want to open it anyway?

1.4 Basic installation

The Ox package is provided in a zip file `excel2ox_900.zip` (or newer). Create a `excel2ox` folder in your `OxMetrics9/ox/packages` folder, and unzip the file into this.

1.5 Overview

The xll installs user-defined functions (UDF) which allow a call from any function from an Ox program. Everytime the function is evaluated by Excel, Ox is started, the function executed, and Ox then terminates.

Because the Ox dynamic-link library is only loaded once in the Excel process, only one Ox based XLL can be active. The XLLs are not re-entrant (i.e. they cannot be used in parallel), which is fine because Excel will not call them asynchronously.

1.6 Argument mapping

From Excel to Ox (function arguments)	
number	double or int
rectangle of numbers	matrix
rectangle some non-numbers	array
string	string
string `<>`	empty matrix
string `{}`	empty array
string `.Inf`	`.Inf`
string `-.Inf`	`-.Inf`
string `+.Inf`	`+.Inf`
string `.NaN`	`.NaN`
string `.Null`	`.Null`
bool	int (0 or 1)
nil	0.0
`#N/A`	`.NaN`
other errors	int (error code: argument error)
From Ox to Excel (function return values)	
int	integer number
double	number
empty matrix	string `"<>"`
1×1 matrix	number
other matrix	rectangle of numbers
string	string
array (1-dimensional)	row of values
array (2-dimensional)	rectangle of values

Dates in Excel are stored as a number that is formatted as a date, so it cannot be distinguished from a regular number. A date in Excel is the number of days after 1899-12-31, except that 1900 is wrongly treated as a leap year. To map to an Ox date, which uses the Julian calendar, you need to add 2415019, provided the date is after February 1900. This value can be computed in Ox as: `dayofcalendar(1899, 12, 31) - 1`. Time is expressed as a fraction of a day, which matches Ox.

2 Excel2Ox

2.1 Installing the Excel2Ox Excel Add-In

The easiest way to start using the Excel2Ox add-in is to open `Excel2Ox-AddIn64-packed.xll` using File/Open in Excel (this will give a warning that the xll is not signed). Alternatively, use the add-in manager: click on the FILE menu, then on Options in the list on the left, then on Add-Ins on the left, and finally on Go to manage Excel Add-ins.

2.2 Functions provided by the Excel2Ox Add-In

The Excel2Ox add-in installs two functions, called `OxRun1`, and `OxRunc`, which differ somewhat in the way arguments for the Ox function are handled.

OxRun1 (*OxFunctionName*, *Arg1*)

1. string with file name, either a `.ox` or a `.oxo` file
   The file should have a path, or be in the same location as the `.xll` file.
2. string with function name
   If this is just a function name then it is taken from `Excel2Ox.ox` that is in the same folder as the xll.
   To specify an Ox file name use: "ox-file?ox-function". For example "test.ox?LogGamma" to use `test.ox` that is in the same folder as the xll. A full path may also be specified
3. first argument for Ox function (optional),

The Ox function can be called as soon as the file and function names are strings, and none of the arguments have an error code.[1] If the number of supplied arguments is wrong, an Ox run-time error will be raised.

OxRunc (*OxFunctionName*, *Argcount*, *Arg1*, ...)

1. string with function name, see OxFunc1.
2. argument count of Ox function,
3. first argument for Ox function,
4. second argument, ...
5. *Argcount*th argument for Ox function.

The Ox function can be called as soon as the file and function names are strings, the correct number of arguments is supplied, and no argument has an error code.[1]

[1] An error code in this case means an Excel error: `#NUM!`, `#REF!`, `#NAME?`, `#VALUE!`, `#DIV/0!`, `#NULL!`, or a string that starts with a `#`. If there is an argument with an error code, the function returns: `#N/A argument error`.
Note that `#N/A` is not treated as an error, but as the Ox missing value `.NaN`.

2.3 Using the Excel2Ox Add-In

The following example uses the `Excel2Ox.ox` file that is part of this package:

```
=OxFunc1("Hello")
```

`Hello` returns a string.

Another example using the `Excel2Ox.ox` file is:

```
=OxFunc1("Test", A3:B4)
```

The `Test` function returns its argument as a string.

The `Sum3` function takes 3 arguments and can be called with `OxFuncc`, where we specify the number of arguments:

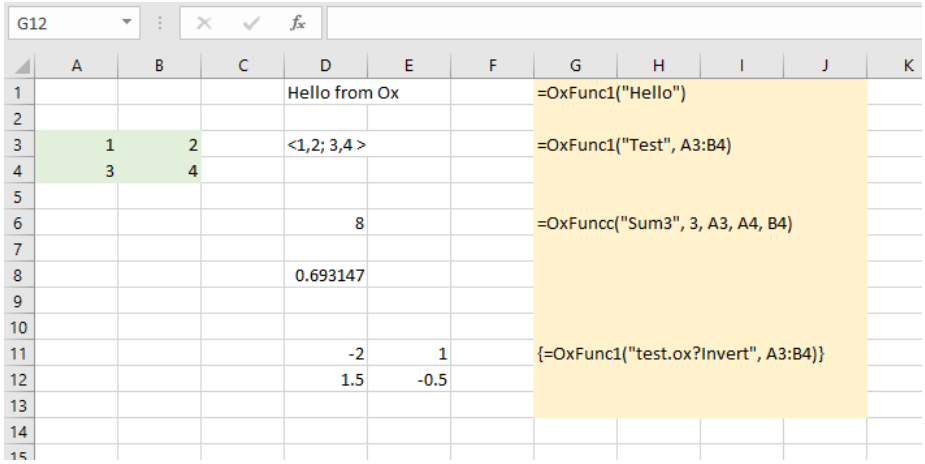
```
=OxFuncc("Sum3", 3, A3, A4, B4)
```

To call the `LogGamma` function from `test.ox`:

```
=OxFunc1("test.ox?LogGamma", A4)
```

`Invert` takes a matrix as input, and returns a matrix. The returned matrix cannot be stored in a single cell. Instead a two by two block should be selected, the formula entered, followed by Ctrl+Shift+Enter to make it in an array expression.

This is the resulting workbook `Test.xlsx`:



2.4 How does it work?

1. Each call to OxFunc1 and OxFuncc works as follows:
   (a) locate the source file, in the folder of the XLL if there is no path, else as specified;
   (b) process the arguments supplied from inside Excel;
   (c) load the Ox code without running it;
   (d) call the specified function;
   (e) exit Ox, and if successful, return the result to Excel;

   Ox run-time errors are shown in a message box.

2.5 Source code

The add-in can call any Ox function without the need to recompile the XLL. The source code of the XLL is in the `Ox_UDF` folder.