Exportxl Macro Tip Sheet

Purpose: The macro exports SAS datasets to Excel workbooks and/or worksheets. It can create new workbooks or worksheets, modify existing worksheets, export to a range that hasn't been predefined, use variable names or labels, use or not use variable formats, use Excel templates or existing worksheets as templates, and create pivot tables.

Named Parameters: The macro uses Named parameters so that: (1) default values can be assigned and (2) the various parameters only have to be specified when values other than the default values are needed.

Parameter	Required	Possible Values	Default	Description
data	Yes	Any valid one or two-level SAS filename	Null	The 1 or2-level filename you want to export
outfile	Yes	Any valid filename (including path) W = provide window for user input Null = path and filename of the data parameter + xlsx extension	Null	The path and filename of the workbook that you want the macro to create or modify
sheet	No	Valid worksheet name W = provide window for user input Null = filename of data parameter	Null	The name of the worksheet you want to create or modify
type	No	P= run PROC EXPORT N = create a new workbook using VBS A = add new worksheet using VBS M = modify worksheet using VBS C = copy dataset to system clipboard	N	The type of process that you want to run
usenames	No	N = don't include a variable name row Y = include a variable name row L = include a variable label row W = provide window for user input	Y	Whether the first row of the range will contain the first data record, the variable names, or the variable labels
range	No	Any valid Excel cell name W = provide window for user input	A1	The upper left cell where you want the table to begin
template	No	Null = No template is to be used filename (including path) of an existing Excel template or workbook W = provide window for user input	Null	The filename (including path) of an Excel template or workbook that you want applied as a template to the file you are exporting
templatesheet	Yes if template specified	Null = No template is to be used Worksheet name W = provide window for user input	Null	The name of the worksheet to be used as a template

useformats	No	Y = Yes
		N = No
		W = provide window for user input
			N	Whether dataset's formats should be applied when exporting its data

usenotepad	No	N = Don't use Notepad
		Y = Use Notepad
			N	Notepad is needed If you're running this macro on a system that doesn't provide direct access to your computer's clipboard (e.g., if you're running this macro on a server)

pivot	No	Null Space separated list of variable names W = provide window for user input	Null	Space separated list of the character variable names to use as Pivot Table's class variables, followed a space, and the name of the analytical variable

Usage Examples: The following examples assume that you have write access to a directory named: c:\temp. Having that specific directory isn't a requirement for the macro, but you do need to have write access to the file that you specify in the outfile parameter

Example 1: Create a new workbook (c:\temp\class.xlsx), copying all records from sashelp.class, letting the macro automatically name the worksheet (i.e., use the data parameter's filename: class), with the worksheet's first row containing the dataset's variable names

```
%exportxl(data=sashelp.class, outfile=c:\temp\class.xlsx)
```

Example 2: Create the same workbook as in Example 1, but name the worksheet 'Students', and don't include a variable name header record

```
%exportxl(data=sashelp.class,  outfile=c:\temp\class.xlsx,usenames=N,  sheet=Students)
```

Example 3: Same as Example 2, but running on a system that doesn't provide direct access to your computer's clipboard (e.g., a server), or have an Excel configuration that clears the clipboard upon opening

```
%exportxl(data=sashelp.class, outfile=c:\temp\class.xlsx, usenames=N, sheet=Students, usenotepad=Y)
```

Example 4: Create a new workbook from sashelp.cars, name the worksheet 'cars', and have the worksheet's first row contain the dataset's variable labels

```
%exportxl( data=sashelp.cars, outfile=c:\temp\cars.xlsx, usenames=L)
```

Example 5: Create a new workbook (c:\temp\class.xlsx), copying all records for males from sashelp.class, name the worksheet 'Males', and have the worksheet's first row contain the dataset's variable names

```
%exportxl( data=sashelp.class(where=(sex eq 'M')), sheet=Males, outfile=c:\temp\class.xlsx)
```

Example 6: Modify the workbook created in Example 5, adding a new worksheet named 'Females', copying all records for females from sashelp.class, and have the worksheet's first row contain the dataset's variable names

```
%exportxl( data=sashelp.class(where=(sex eq 'F')), sheet=Females, outfile=c:\temp\class.xlsx, type=A)
```

Example 7: Create a workbook using an Excel template

```
%exportxl( data=sashelp.class (keep=name sex age height), template=c:\temp\template.xltx,
     templatesheet=template, outfile=c:\temp\class_stats.xlsx, usenames=N, range=A2, sheet=Jan_2018)
```

Example 8: Modify workbook created by running Example 7, adding the weight variable to column E

```
%exportxl( data=sashelp.class (keep=weight), type=M, range=E2, outfile=c:\temp\class_stats.xlsx,
    usenames=N, sheet=Jan_2018)
```

Example 9: Create a new workbook including a Pivot Table

```
%exportxl( data=sashelp.cars, outfile=c:\temp\cars.xlsx, pivot=Origin Type Make MSRP)
```