



---

# Choosing the Right Tool from Your SAS<sup>®</sup> and Microsoft Excel<sup>®</sup> Tool Belt



**SYSTEMS SEMINAR CONSULTANTS, INC.**

2997 Yarmouth Greenway Drive, Madison, WI 53711

Phone: (608) 278-9964 • Web: [www.sys-seminar.com](http://www.sys-seminar.com)



---

# Choosing the Right Tool from Your SAS<sup>®</sup> and Microsoft Excel<sup>®</sup> Tool Belt

This paper was written by Systems Seminar Consultants, Inc. SSC specializes in SAS software and offers:

- Training Services
- Consulting Services
- Help Desk Plans
- Newsletter Subscriptions to *The Missing Semicolon*<sup>™</sup>

COPYRIGHT© 2010 Systems Seminar Consultants, Inc.

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without prior written permission of SSC. SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. *The Missing Semicolon* is a trademark of Systems Seminar Consultants, Inc.

# Free SAS Newsletter

---



The Missing Semicolon™, shares SAS software solutions developed by our staff and provides additional technical assistance to our customers.

Visit [www.sys-seminar.com](http://www.sys-seminar.com) to sign up for a free subscription:





**Steven J. First**  
**President**

- Over 30 years of SAS experience, including hundreds of manufacturing, retail, government, marketing, and financial applications.
- Over 25 years as President and Founder of SSC
- Founder of WISAS and WISUG
- Invited speaker at local, regional, and international user groups



**Jennifer First**  
**Director of Operations**

- Directs SSC's administrative, sales, and IT staff.
- Strategic business planning.
- Assists clients with project planning, customized training
- Coordinates Wisconsin SAS User Group Meetings.
- Publishes The Missing Semicolon, SSC's SAS technical newsletter.

# Introduction

---



- SAS software analytics, reporting, and programming are second to none.
- Excel is probably the most popular software product used today.
- Excel is easy and flexible to users.
- Excel truly runs many parts of business.
- SAS and Excel together make good software into great software.



© Scott Adams, Inc./Dist. by UFS, Inc.

# Sending Excel to SAS

---



- Most people are familiar with Excel.
- Excel provides easy navigation.
- Easy data input.
- Formulas, formatting, much more available.
- Contents can be a rich input source for SAS.

# Sending SAS to Excel

---



- Excel is de-facto report distribution system.
- Can contain reports, graphs, and other SAS results in a familiar format.
- Can also contain templates, formulas.
- Client can alter as they see fit.

# Some Questions about SAS and Excel

---



- Are the structures similar?
- How to transfer from SAS files to Excel and vice versa?
- Are there formatting differences?
- What is the best import/export method for your project?

# What Are SAS Capabilities?

---



A Small Subset:

- SAS is excellent for analytics after data has been captured.
- SAS is somewhat batch oriented.
- SAS may require programming depending on which products are used.
- SAS has extensive programming and reporting capabilities.
- Excellent data manipulation tool.

# What About Excel Capabilities?

---



- Excel is a much more interactive and simple to use tool.
- It is very easy to use and learn.
- Users can begin using Excel almost immediately.
- Putting data and formulas into cells is straight forward.
- Excel is an almost universal format for exchanging reports, worksheets.
- Most users don't do much programming in Excel.
- Macro language, interfaces to several programming languages available.

# Are SAS and Excel Structures Similar?

---



Shapes of SAS, Excel data usually close enough to make transfer easy:

- SAS datafiles, Excel worksheets are both rectangles (rows and columns).
- Data values are stored in similar manners.
- Formatting is similar.
- Date handling is similar.

# What Are The SAS and Excel Structure Differences?

---



## SAS:

- has a required dataset descriptor( mini data dictionary) defining columns.
- columns are named uniquely.
- columns must contain all numeric or all character values.
- names are more restrictive.
- cannot reference space, \$ etc. in names without special naming.

## Excel:

- does not require columns to be named.
- columns can contain a mixture of character, numeric values.
- can also contain formulas.

# Do SAS and Excel Use “Libraries”?

---



## SAS

- Stores its data in SAS libraries.
- Libraries are roughly the equivalent of a Windows or Unix directory.
- Each SAS file is a separate Windows or Unix file.
- Similar techniques on mainframes.
- LIBNAME statement normally names a SAS library.

## Excel

- Is somewhat of a library.
- Workbooks are a single file.
- Can be considered a library.
- Each range being a separate member inside it.

Defining libraries, members this way should ease transfer between SAS and Excel.

# How Can SAS Send Information to Excel?

---



Before we can do any transfer, are we talking about:

- A SAS dataset?

Or

- A SAS report?

We can do both!!

# How About Sending a SAS Report to Excel?

---



In the past:

- PROC PRINTTO could capture print.
- Separately we could parse the report to create suitable input for Excel.

Now with ODS:

- Process has become much simpler.
- ODS can capture any SAS report.
- ODS can route it to various destinations.
- One of the simplest files to create is an HTML file.
- HTML is normally used to create web pages.
- Excel does an almost automatic process to convert HTML to Excel format.
- This process becomes trivial.

# Can You Show us an Example?

---



Suppose we write the following proc tabulate code to produce a report:

```
proc tabulate data=softsale;  
title 'Softsale Sales and Expenses by Division';  
class state division ;  
var sales expense;  
table state,division*sales*sum  
          division*expense*sum;  
run;
```

# Can We See the Output?



Softsale Sales and Expenses by Division

	Division		Division	
	H	S	H	S
	Sales	Sales	Expense	Expense
	Sum	Sum	Sum	Sum
State				
IL	4009.21	743.22	322.12	159.45
MN	3231.75	7732.44	644.55	1339.45
WI	12185.10	8232.11	2786.10	3339.41

# So How Do We Get it to Excel?

---



We can easily route the output to HTML with a few ODS statements.

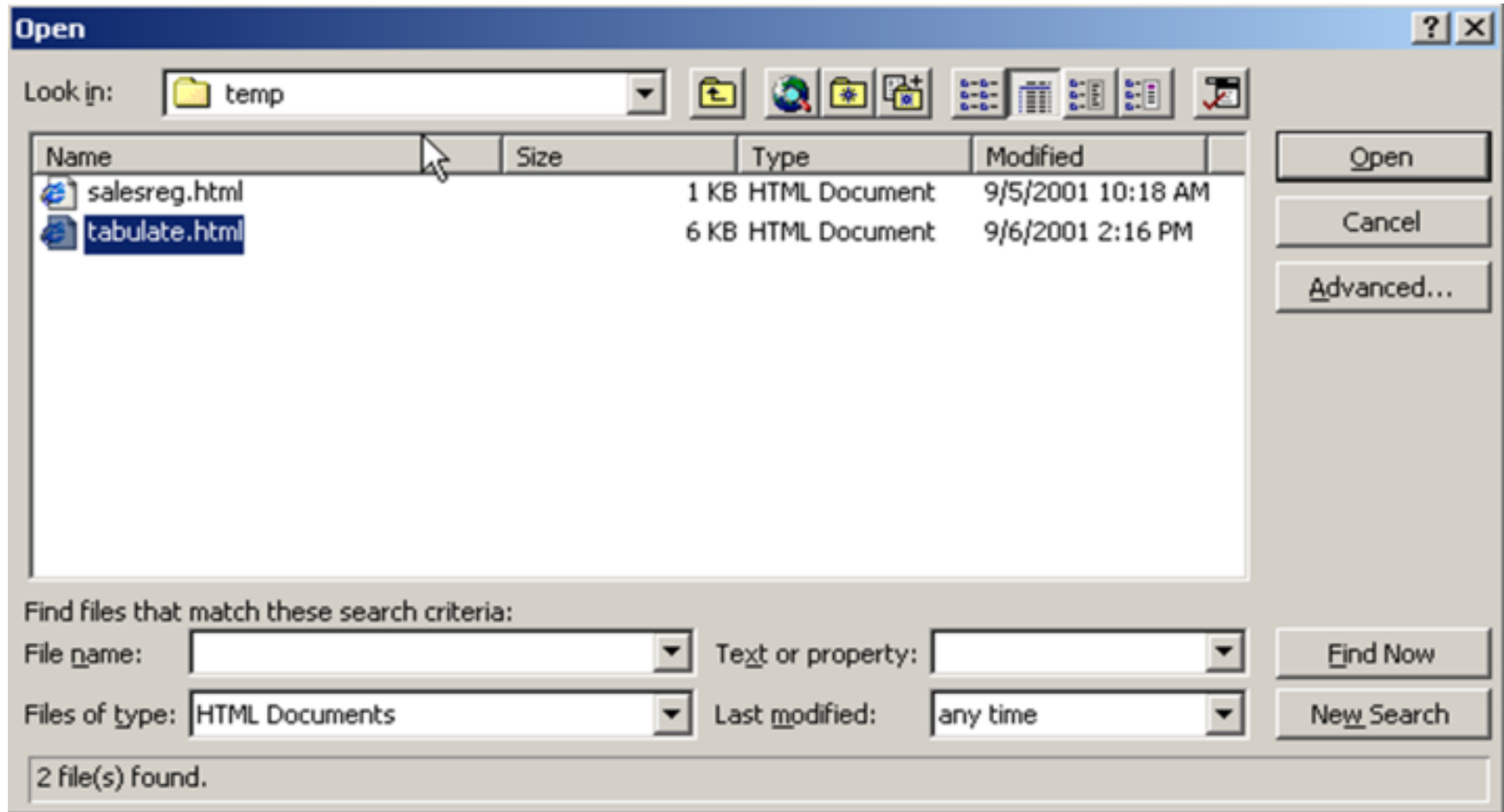
```
ods html body='c:\temp\tabulate.html';  
proc tabulate data=softsale;  
  . . .  
run;  
ods html close;
```

We now have a file that can be published on a Web site.

# How Does That Help Us Get to Excel?



Easy! To open the file in Excel, use **File Open** and specify file type HTML.



# What Does Excel Do with the HTML File?



- Excel automatically converts HTML as it reads the file.
- The user may have to specify to save it as an XLS format.

Microsoft Excel - tabulate2.html

File Edit View Insert Format Tools Data Window Help

Type a question for help

A2 fx

Softsale Sales and Expenses by Division

		Division		Division	
		H	S	H	S
		Sales	Sales	Expense	Expense
		Sum	Sum	Sum	Sum
State		4009.21	743.22	322.12	159.45
IL					
MN		3231.75	7732.44	644.55	1339.45
WI		12185.1	8232.11	2786.1	3339.41

tabulate

Ready NUM

# Can We Make This Process More Transparent?

---



Yes!

- Continue to generate HTML but naming our file with .XLS.
- Excel and Windows are tricked to think they are getting an Excel file.
- The file will be converted automatically upon input.
- The user can double click on the file to load Excel.
- If color and styles aren't critical, use style=minimal (reduces file size significantly).

# Can We See the Code?



options nocenter;

ods html body='c:\temp\tabulate.xls' style=minimal;

proc tabulate data=softsale;

...

run;

ods html close;

Microsoft Excel - tabulate.xls

File Edit View Insert Format Tools Data Window Help

A3 fx

Softsale Sales and Expenses by Division					
	Division		Division		
	H	S	H	S	
	Sales	Sales	Expense	Expense	
	Sum	Sum	Sum	Sum	
State	4009.21	743.22	322.12	159.45	
IL					
MN	3231.75	7732.44	644.55	1339.45	
WI	12185.1	8232.11	2786.1	3339.41	

tabulate

Ready NUM

# Are There Differences in Formatting?

---



Excel may format things differently than SAS does:

- Leading zeros that don't appear in Excel.
- Text doesn't span columns in the worksheet quite the way you like it to.

Can we do anything about that?

- In our SAS code, we actually can program an Excel formula.
- We can control HTML spanning.
- Excel may have formatting commands to handle these problems.

Examples:

```
Account='="00003444";
```

```
Title '<td colspan=5> My Title </td>';
```

# Is There a More Controlled Way to Send Data to Excel?

---



## ODS Markup Family of Destinations:

- Designed to create markup codes for easy input by other software.
- The ODS specifies MARKUP along with TAGSET= option.
- Alternate form is to specify the TAGSET.tagset as ODS destination.
- Tagsets exist for XML, CSV, EXCELXP, and many more.  
(EXCELXP designed exclusively for input to Excel).
- EXCELXP has many more options to control almost anything in the spreadsheet: such as styles, titling, footnotes, by lines, worksheet names. and much more.

# Are the Tagsets Documented?

---



- SAS.COM
- Online Help
- Run the following Program:

```
ods tagsets.excelxp file='test.xml' options(doc='help');
```

# Can We See an Example Using EXCELXP?

---



An example of a simple PROC PRINT using EXCELXP follows:

```
ods tagsets.excelxp file='c:\temp\test.xml';  
proc print data=softsale(obs=5) noobs;  
  var Name Division Years Sales Expense State;  
  title 'Softsale First 5 Obs';  
run;  
ods tagsets.excelxp close;
```

# What Does The Worksheet Look Like?



- The Worksheet looks like the report.
- Titles don't appear, as they are instead used as print headers.
- SAS will generate the worksheet names.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - test.xml". The worksheet contains a table with the following data:

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Division	Years	Sales	Expense	State					
2	BENJAMIN	S	3	201.11	25.21	IL					
3	JENNIFER	S	1	542.11	134.24	IL					
4	PAT	H	4	4009.21	322.12	IL					
5	ANDREW	S	24	1762.11	476.13	MN					
6	SARAH	S	6	301.21	65.17	MN					
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											

# Can I Include Titles and Specify the Sheet Name?

---



Options are available:

- Set the worksheet name prefix.
- SAS will use to create a separate worksheet for each by group.
- Titles (if any), are now imbedded in the worksheet itself.
- Many, many more options.

```
ods tagsets.excelxp file='c:\temp\test.xml'  
  options(embedded_titles='Yes' Sheet_Name='State');  
proc print data=softsale(obs=5) noobs;  
  var Name Division Years Sales Expense State;  
  by state;  
  title 'Softsale First 5 Obs';  
run;  
ods tagsets.excelxp close;
```

# The Resulting Worksheet



Titles are imbedded and each state will be in a separate tab.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - test.xml". The worksheet is named "Softsale First 5 Obs". The data is as follows:

Softsale First 5 Obs					
State=IL					
Name	Division	Years	Sales	Expense	State
BENJAMIN	S	3	201.11	25.21	IL
JENNIFER	S	1	542.11	134.24	IL
PAT	H	4	4009.21	322.12	IL

# While Reporting is Great, Can We Copy *Data*?

---



- There are a wealth of ways to copy SAS data to Excel.
- We can eliminate the reporting step.
- Some require only Base SAS or Enterprise Guide.

# Can We Copy and Paste From SAS?

---



- Copying and pasting all the cells from a SAS dataset in the SAS VIEWTABLE window is not supported.
- A single cell can be copied.

We have 2 quick options:

- Copy and paste from The SAS System Viewer.
- Use Enterprise Guide to copy and paste.

Notes:

We can also write complex SCL code for cut and paste.

# What is the SAS System Viewer?

---



- A free Windows application to view SAS data.
- Download it from The SAS Institute at:

[http://www.sas.com/apps/demosdownloads/92 SDL\\_sysdep.jsp?packageID=000512](http://www.sas.com/apps/demosdownloads/92 SDL_sysdep.jsp?packageID=000512).

# Can You Show Copy and Paste In the System Viewer?



In the SAS System Viewer:

- Open the data set you want to send to Excel.
- Select all of the rows and columns you want (or the whole dataset).
- Use the Windows shortcut to copy (Ctrl+C) (no Edit menu).

SAS Universal Viewer - [C:\temp]

File Tools Help

Address

Tables SOFTSALE

Table

Freeze Hide Show... Format Filter...

	Name	Division	Years	Sales	Expense	State
1	CHRIS	H	2	233.11	94.12	WI
2	MARK	H	5	298.12	52.65	WI
3	SARAH	S	6	301.21	65.17	MN
4	PAT	H	4	4009.21	322.12	IL
5	JOHN	H	7	678.43	150.11	WI
6	WILLIAM	H	11	3231.75	644.55	MN
7	ANDREW	S	24	1762.11	476.13	MN
8	BENJAMIN	S	3	201.11	25.21	IL
9	JANET	S	1	98.11	125.32	WI
10	STEVE	H	21	6153.32	1507.12	WI
11	JENNIFER	S	1	542.11	134.24	IL
12	JOY	S	12	2442.22	761.98	WI
13	MARY	S	14	5691.78	2452.11	WI
14	TOM	S	5	5669.12	798.15	MN
15	BETH	H	12	4822.12	982.1	WI

Rows 1-15 of 15 Filter: none Sort: none 6,15

C:\temp

initializing...

# Can We then Paste Into Excel?



In Excel:

- Use the Windows shortcut to paste (Ctrl+V).
- You should be able to paste all of your data into Excel.
- You may not be able to paste in variable names.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	CHRIS	H		2	233.11	94.12	WI								
2	MARK	H		5	298.12	52.65	WI								
3	SARAH	S		6	301.21	65.17	MN								
4	PAT	H		4	4009.21	322.12	IL								
5	JOHN	H		7	678.43	150.11	WI								
6	WILLIAM	H		11	3231.75	644.55	MN								
7	ANDREW	S		24	1762.11	476.13	MN								
8	BENJAMIN	S		3	201.11	25.21	IL								
9	JANET	S		1	98.11	125.32	WI								
10	STEVE	H		21	6153.32	1507.12	WI								
11	JENNIFER	S		1	542.11	134.24	IL								
12	JOY	S		12	2442.22	761.98	WI								
13	MARY	S		14	5691.78	2452.11	WI								
14	TOM	S		5	5669.12	798.15	MN								
15	BETH	H		12	4822.12	982.1	WI								
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															

# Can You Copy and Paste From Enterprise Guide?



In Enterprise Guide:

- Open your dataset.
- Highlight all of the rows and columns you want.
- Edit menu and select copy (CTRL C).

The screenshot shows the SAS Enterprise Guide interface. The main window displays a dataset named 'SOFTSALE (read-only)' with the following data:

	Name	Division	Years	Sales	Expense	State
1	BENJAMIN	S	3	201.11	25.21	IL
2	JENNIFER	S	1	542.11	134.24	IL
3	PAT	H	4	4009.21	322.12	IL
4	ANDREW	S	24	1762.11	476.13	MN
5	SARAH	S	6	301.21	65.17	MN
6	TOM	S	5	5669.12	798.15	MN
7	WILLIAM	H	11	3231.75	644.55	MN
8	BETH	H	12	4822.12	982.1	WI
9	CHRIS	H	2	233.11	94.12	WI
10	JANET	S	1	98.11	125.32	WI
11	JOHN	H	7	678.43	150.11	WI
12	JOY	S	12	2442.22	761.98	WI
13	MARK	H	5	298.12	52.65	WI
14	MARY	S	14	5691.78	2452.11	WI
15	STEVE	H	21	6153.32	1507.12	WI

In Excel:

- Edit menu and select paste(CTRL V).

# Can Data Step Programs Send SAS Data to Excel?

---



A data step can create a comma separated value (CSV) text file.

Features:

- Wide capabilities of the data step make this a very flexible tool.
- DSD FILE option inserts a delimiter (,) between values.
- DSD quotes the values if they contain special characters.
- The Tilde (~) format modifier will quote all fields.
- ODS CSV destination with Proc Print gives virtually identical results.
- When opened in Excel, it is automatically parsed into columns.

# Are There Drawbacks to a Data Step Program?

---



- A Data step must be written.
- You need know names of SAS columns for header rows.
- You must also specify column names to change column order.
- The program must alter the data step for each dataset used.
- Manual steps are needed in Excel.

# Can We See an Example?

---



The following program creates a CSV file:

```
data _null_;  
  set softsale(obs=5);  
  file 'c:\temp\test.csv' dsd;  
  if _n_ =1 then  
    put 'Name,Division,Years,Sales,Expense,State';  
  put ( _all_ ) (+0);  
run;
```

# What Does the CSV File Look Like?

---



File 'c:\temp\test.csv'

```
Name,Division,Years,Sales,Expense,State  
BENJAMIN,S,3,201.11,25.21,IL  
JENNIFER,S,1,542.11,134.24,IL  
PAT,H,4,4009.21,322.12,IL  
ANDREW,S,24,1762.11,476.13,MN  
SARAH,S,6,301.21,65.17,MN
```

# How Does It Look In Excel?



Here is the Excel view after opening the above file.

The screenshot shows the Microsoft Excel interface with the file 'test.csv' open. The data is displayed in a table with the following columns: Name, Division, Years, Sales, Expense, and State. The rows contain employee information.

	A	B	C	D	E	F	G	H	I
1	Name	Division	Years	Sales	Expense	State			
2	BENJAMIN	S	3	201.11	25.21	IL			
3	JENNIFER	S	1	542.11	134.24	IL			
4	PAT	H	4	4009.21	322.12	IL			
5	ANDREW	S	24	1762.11	476.13	MN			
6	SARAH	S	6	301.21	65.17	MN			
7									

# Wouldn't This Be a Great Application for a Macro?

---



There are many user written macros to create CSV files.

- %SSCFLAT macro reads dictionary tables to gather SAS Info.
- Produces the CSV file with minimum user coding.
- Runs on all platforms.
- Options exclude header rows, use column names or labels.
- Many other options.
- SAS Inc. macro %ds2csv very similar.
- Both macros are free.

# What is DDE?

---



Dynamic Data Exchange (DDE) is a method of dynamically exchanging information between Windows applications.

- Almost any command, input to Excel can be done by data step, DDE.
- Very powerful logic using the best of the DATA step, along with Excel.
- DDE has existed for many years, mature, well documented.
- Uses a client/server relationship to enable a client application to request information from a server application.
- SAS is always the client.
- Harnesses the power of the data step while interfacing with MS Office.

Notes:

- Only Base SAS and Microsoft Office need be licensed.

# What is Needed to Start a DDE Server?

---



- Excel must have an open worksheet before running the SAS code.
- A DDE server app can be opened using the X command within SAS.
- The XWAIT and XSYNC options must be turned off.

```
options noxwait noxsync;  
x "c:\microsoft office\office\excel.exe";
```

# Can We See a Data Step That Writes to DDE?

---



Example: send first 5 rows and selected columns from a SAS Excel.

```
filename excelout dde 'excel|sheet1!r1c1:r5c6';  
data _null_;  
  set softsale(obs=5);  
  file excelout;  
  put Name Division Years Sales Expense State;  
run;
```

# What Do the Results Look Like?



A Worksheet in Excel.

	A	B	C	D	E	F	G	H
1	BENJAMIN	S	3	201.11	25.21	IL		
2	JENNIFER	S	1	542.11	134.24	IL		
3	PAT	H	4	4009.21	322.12	IL		
4	ANDREW	S	24	1762.11	476.13	MN		
5	SARAH	S	6	301.21	65.17	MN		
6								

# What is the SAS/Access Interface to PC File Formats?

---



With this additionally licensed product, transfer of data becomes even easier.

The product contains multiple methods for converting data:

- Export/Import wizards to convert data to and from multiple file formats.
- PROC EXPORT/IMPORT code can be captured, run through SAS.
- Excel Libname engine treats Excel workbooks as SAS libraries.
- PROC SQL pass-through facility can also treat Excel worksheets as SAS datasets.
- PROC ACCESS to create access and view descriptors.
- PROC DBLOAD to load Excel worksheets.

Notes:

- PROC ACCESS, DBLOAD are generally older included mostly for compatibility.
- They may have other SAS 6 limitations, (8-byte SAS variable names).

# How Does The Export Wizard Work?

---

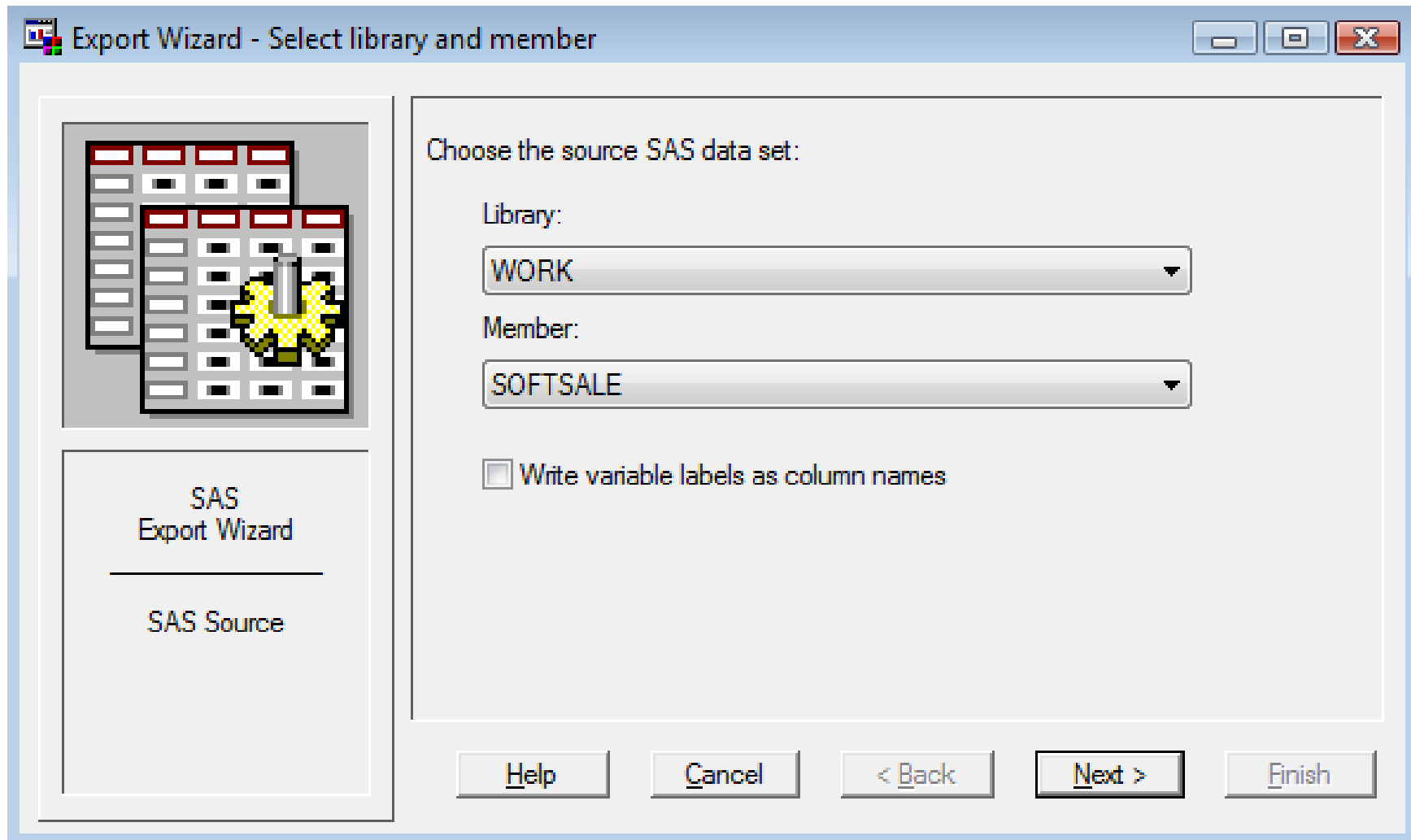


- A point-and-click interface.
- Guides you through reading SAS data, writing to an external file format.
- External file formats can include many other formats besides Excel.
- We just follow instructions and click.
- If we need help, we can click the Help button.
- PROC EXPORT code is generated and run, and we can save it as well.

# How Do You Start the Export Wizard?



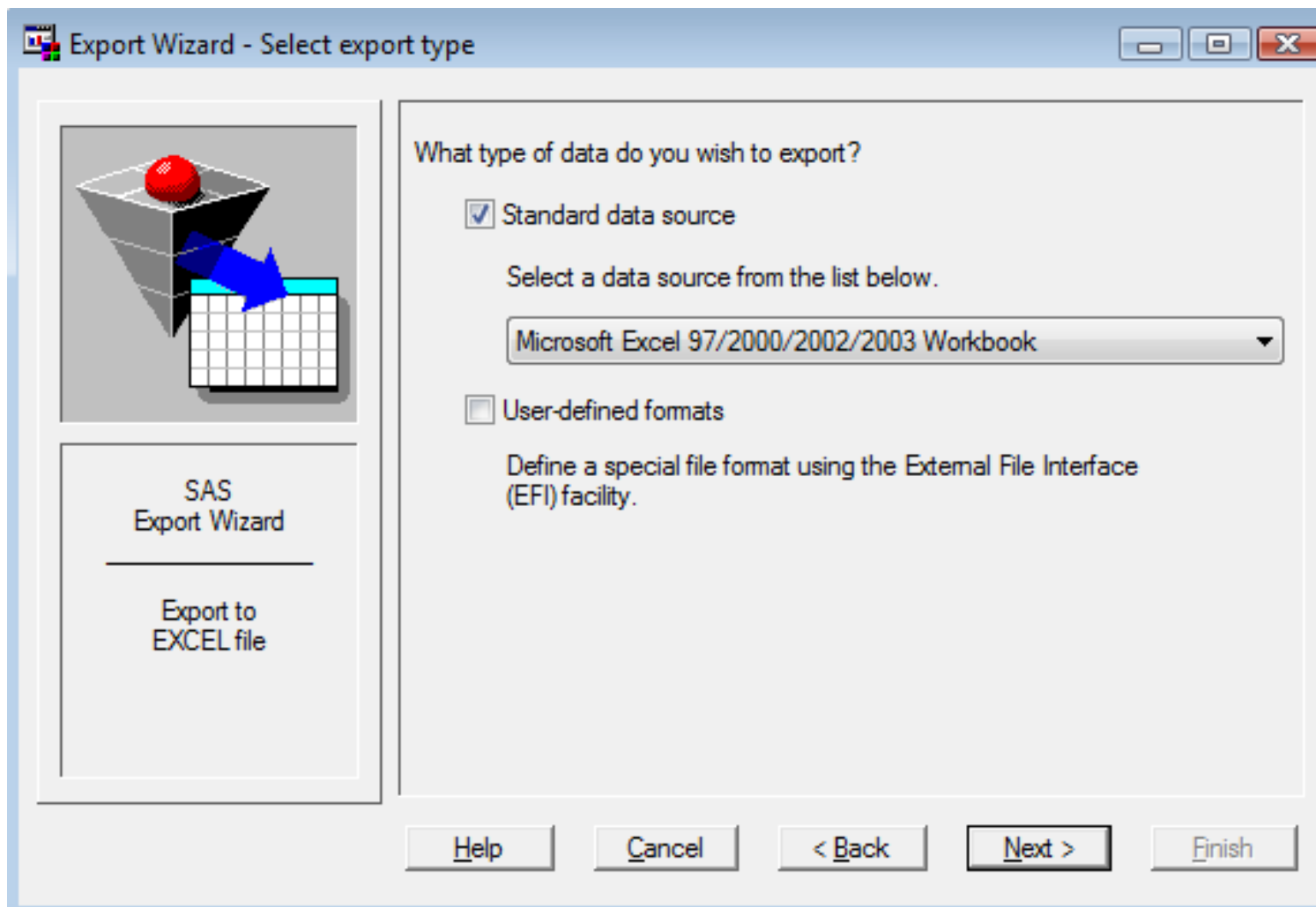
- Open the Export Wizard, selecting File, Export Data, Select Library.
- Member window opens, select Library and Member, then Next.



# And Then What Happens?



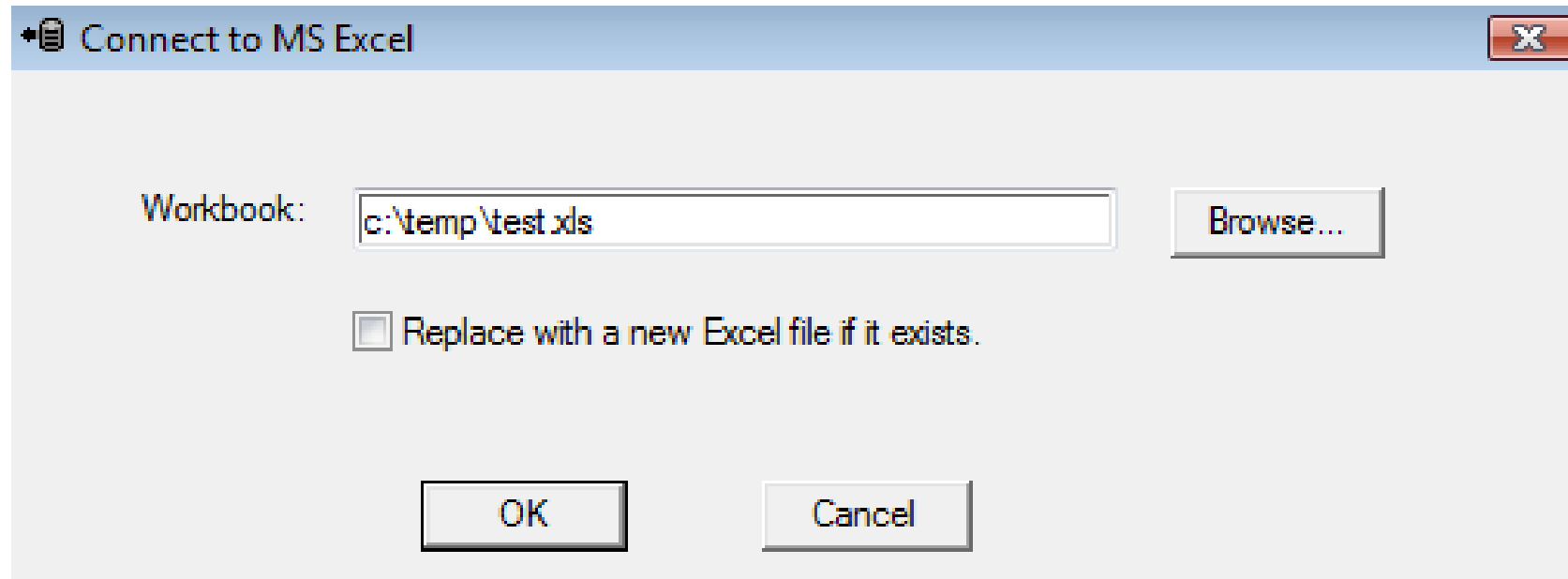
- The Select Export Type window opens.
- Select Export Type in the Export Wizard from the drop down list, Next.



# Where Do We Give an Excel Workbook Name?



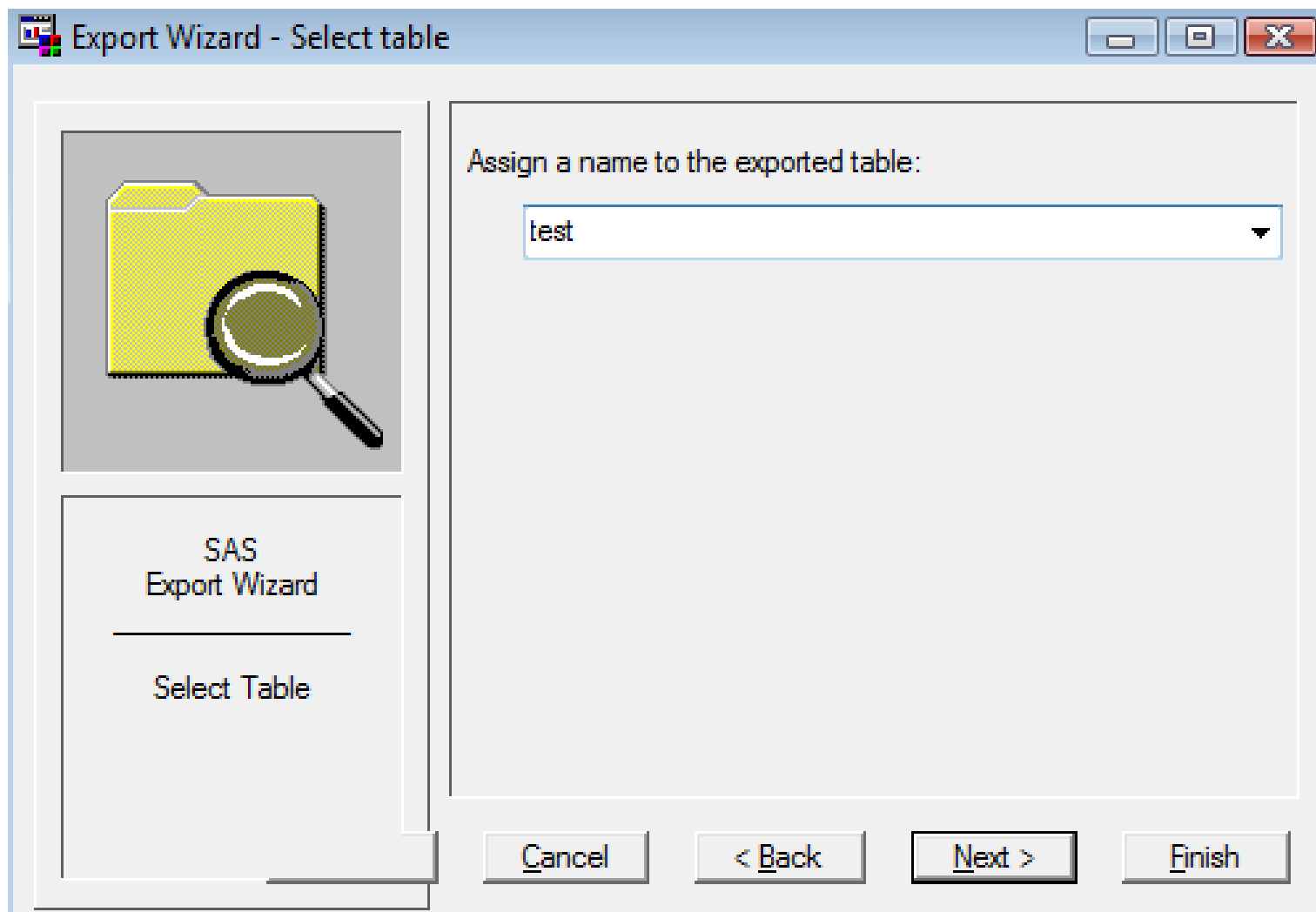
- The Connect to MS Excel dialog box opens, type or Browse as needed, then OK.



# Can We Specify a Table Name (Named Range)?



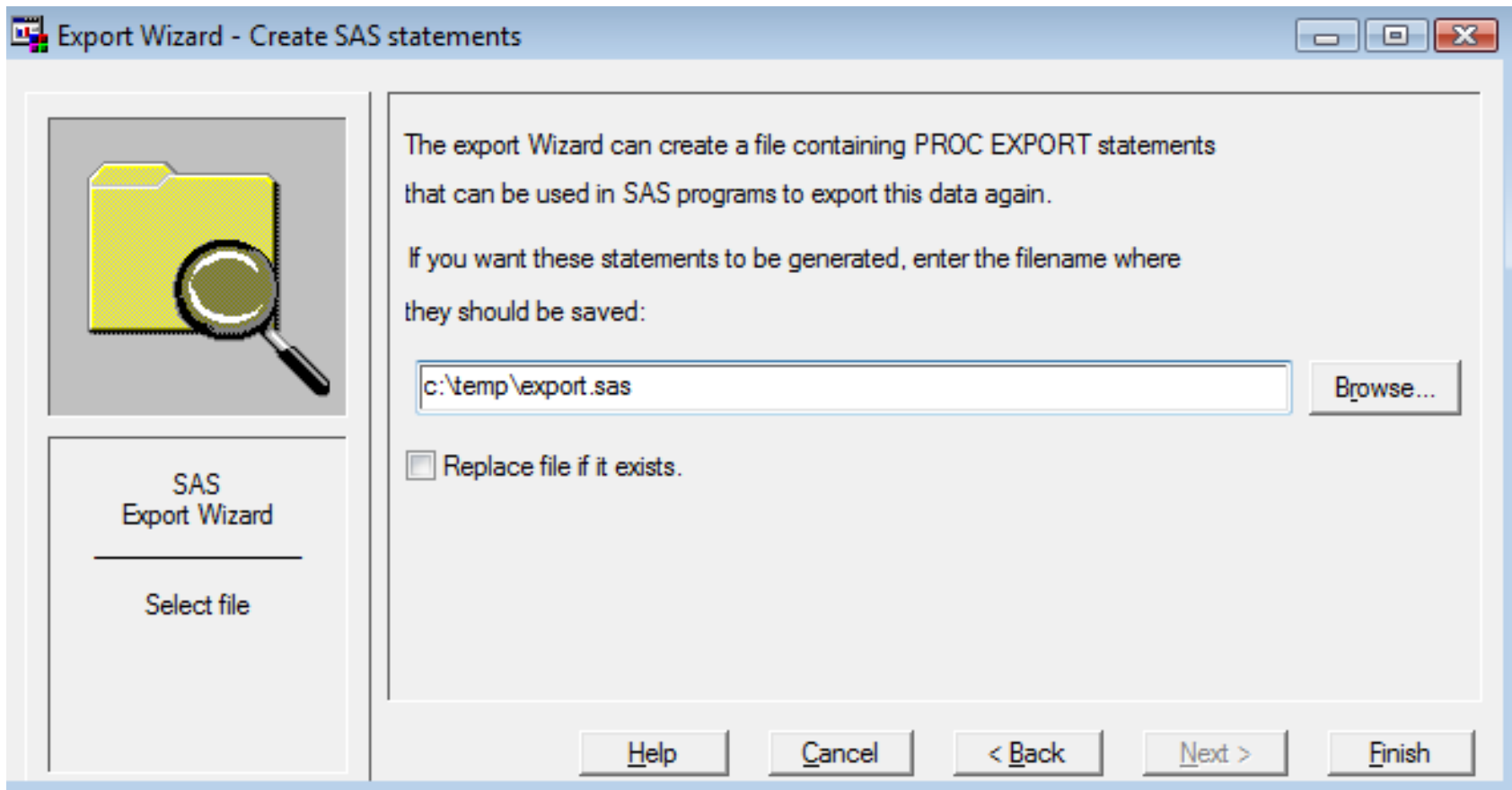
The Select Table window allows a name to be typed or pulled down.



# Can You Save the Generated SAS Code?



- The Create SAS statements window opens.
- We can pull down or type a file for storage and later running of SAS code.
- Click Finish.



# Can We See the Resulting Worksheet?



The worksheet looks almost identical to our SAS dataset.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - test.xls". The worksheet contains the following data:

	A	B	C	D	E	F	G	H	I	J
1	Name	Division	Years	Sales	Expense	State				
2	BENJAMIN	S	3	201.11	25.21	IL				
3	JENNIFER	S	1	542.11	134.24	IL				
4	PAT	H	4	4009.21	322.12	IL				
5	ANDREW	S	24	1762.11	476.13	MN				
6	SARAH	S	6	301.21	65.17	MN				
7	TOM	S	5	5669.12	798.15	MN				
8	WILLIAM	H	11	3231.75	644.55	MN				
9	BETH	H	12	4822.12	982.1	WI				
10	CHRIS	H	2	233.11	94.12	WI				
11	JANET	C	1	88.11	125.23	WI				

Notes:

- Many users think this is the easiest way to send SAS data to Excel.

# What If I Like to Code, or Need Some Special Options?

---



The Wizards either:

- Generated DATA step code.
- Generated SAS/ACCESS code.
- Generated code using a translation engine.

By capturing it or coding from scratch gives repeatability and many more options.

# Can We See the Code From Our Last Example?

---



Here are the statements and arguments saved by the wizard:

```
PROC EXPORT DATA= WORK.SOFTSALE  
    OUTFILE= "c:\temp\test.xls"  
    DBMS=EXCEL REPLACE;  
    RANGE="test";  
RUN;
```

Options are available to:

- Use SAS labels for column headers.
- Specify output sheet names, much more.

SAS docs are very thorough and thus won't be repeated here.

# Advantages/Disadvantages of PROC EXPORT?

---



## Advantages:

- PROC EXPORT is a straightforward conversion.
- It has many options.
- It is repeatable.

## Disadvantages:

- Wizard and/or coding is required.
- A copy of the data is made.

# Is There a LIBNAME Engine to Excel?

---



SAS/ACCESS for PC file formats does support LIBNAME.

- Probably the most simple of all techniques.
- LIBNAME assigns a libref to Microsoft Excel and other data sources.
- Allows you to reference an Excel workbook as a SAS library.
- Worksheets within the workbook are treated as SAS data files.
- Some naming conventions and internal storage are different.
- Generally we can read and write Excel data as a SAS dataset.

# What is a “Table” in the Excel Library?

---



The SAS documentation deals with this topic as follows:

- SAS/ACCESS treats an Excel workbook as a database.
- It treats a *named-range* as a table.
- The named-range must be defined in the Excel file before it can be used by SAS.
- A *worksheet* is treated as a special range (worksheet name with \$).

# Can We See an Example?

---



For example:

- Sheet1 is a sheet name in an Excel file.
- SAS treats Sheet1\$ as a valid range name.
- SAS must use special naming because of the \$ ( 'Sheet1\$'n ).
- First row in a range is normally treated as a column name and is used for SAS variables names.

```
libname myxls excel path="c:\temp\test.xls";  
proc print data=myxls.'sheet1$'n;  
run;
```

# What Can The LIBNAME Excel Engine Do?

---



- Create new workbooks.
- Create a new worksheet within a named range, write data to that range.
- Write data to an empty existing named range.
- Append data to worksheet data or a named range.
- Read data from existing worksheets and ranges.
- Delete all the data in a worksheet or range.
- Do all the above without Excel installed.

# What Can The LIBNAME Excel Engine Not Do?

---



- Rename worksheets in a workbook.
- Delete worksheets or workbooks.
- Change or apply formatting.
- Delete cells with formulas.
- Write formulas into a cell.

## Notes:

- The SAS x command and others can rename, delete workbooks.

# Can Libname Write to a Worksheet?

---



Libname can write to a worksheet.

Example:

- Write out a worksheet named “test”.
- SAS will create a named range called “test\$”.
- Treat TEST like any other SAS dataset.

```
libname myxls excel path="c:\temp\test.xls";  
data myxls.test;  
  set softsale;  
run;  
proc print data=myxls.test;  
run;  
libname myxls clear;
```

# Can LIBNAME Overwrite an Existing Worksheet?

---



If the program is run again it would fail as the worksheet exists.  
We can precede it with a delete statement.

```
options noxwait;  
x 'del c:\temp\test.xls' noxwait;  
libname myxls excel path="c:\temp\test.xls";  
data myxls.test;  
  set softsale;  
run;  
proc print data=myxls.test;  
run;  
libname myxls clear;
```

# Can PROC SQL Write to a Worksheet?

---



PROC SQL writes to a worksheet via CREATE TABLE.

```
options noxwait;  
x 'del c:\temp\test.xls' noxwait;  
libname myxls excel path="c:\temp\test.xls";
```

```
proc sql;  
create table myxls.test as  
select * from softsale;  
quit;
```

```
libname myxls clear;
```

# Can Base SAS *Read* Excel Data?

---



- SAS can also read data from Excel with a variety of methods.
- Many almost identical (but opposite) techniques as already discussed.

# Can We Copy and Paste to SAS DATALINES?

---



- COPY and Paste is about as simple as it gets.
- You can copy cells from Excel, paste after a DATALINES statement.
- Paste does not line the columns up (may be a way around).
- Perhaps use List input in our data step.
- Or we could insert spaces to line up the data in columns.

## Advantages:

- Extremely simple.

## Disadvantages:

- We have to code a data step.
- Would only work for small files.
- Columns may not line up.

# Can We See An Example?



In Excel highlight the desired cells and issue the edit copy command (Ctrl C).

The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - test.xml'. The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains various icons for file operations and editing. The spreadsheet has columns A through F and rows 5 through 16. The active cell is A2, which contains the name 'BENJAMIN'. The data in the spreadsheet is as follows:

	A	B	C	D	E	F
5	ANDREW	S	24	1762.11	476.13	MN
6	SARAH	S	6	301.21	65.17	MN
7	TOM	S	5	5669.12	798.15	MN
8	WILLIAM	H	11	3231.75	644.55	MN
9	BETH	H	12	4822.12	982.1	WI
10	CHRIS	H	2	233.11	94.12	WI
11	JANET	S	1	98.11	125.32	WI
12	JOHN	H	7	678.43	150.11	WI
13	JOY	S	12	2442.22	761.98	WI
14	MARK	H	5	298.12	52.65	WI
15	MARY	S	14	5691.78	2452.11	WI
16	STEVE	H	21	6153.32	1507.12	WI

The status bar at the bottom shows 'Ready', 'Sum=44852.91', and 'NUM'.

# And The Paste Command in SAS

---



```
data softsale;
  input Name $ Division $ Years Sales Expense State $;
  datalines;
BENJAMIN S 3 201.11 25.21 IL
JENNIFER S 1 542.11 134.24 IL
PAT H 4 4009.21 322.12 IL
ANDREW S 24 1762.11 476.13 MN
...
MARK H 5 298.12 52.65 WI
MARY S 14 5691.78 2452.11 WI
STEVE H 21 6153.32 1507.12 WI
;
run;
proc print data=softsale;
run;
```

# Can The DATA Step Read Directly From the Clipboard?

---



The FILENAME CLIPBRD access reads directly from the Windows Clipboard (we can skip the paste step).

```
filename myexcel clipbrd;  
data softsale;  
infile myexcel firstobs=2;  
  input Name $ Division $ Years Sales Expense State $;  
run;  
proc print data=softsale;  
run;
```

# Can Excel Create a CSV File?

---



EXCEL's File Save As, choosing a file type of CSV.

```
Name,Division,Years,Sales,Expense,State
```

```
BENJAMIN,S,3,201.11,25.21,IL
```

```
JENNIFER,S,1,542.11,134.24,IL
```

```
PAT,H,4,4009.21,322.12,IL
```

```
ANDREW,S,24,1762.11,476.13,MN
```

```
...
```

```
MARY,S,14,5691.78,2452.11,WI
```

```
STEVE,H,21,6153.32,1507.12,WI
```

# Can SAS Read the CSV File?

---



- The DSD INFILE option separates at the commas.
- FIRSTOBS= option can ignore the header line.

```
data softsale;
  infile 'c:\temp\test.csv' dsd firstobs=2;
  input Name $ Division $ Years Sales Expense State $;
run;
proc print data=softsale;
run;
```

## Advantages:

- DSD does most of the work.
- and only Base SAS is needed.

## Disadvantages:

- We have to code, specify column names, types, and lengths.

# Can DDE And XML Read EXCEL DATA Within SAS?

---



- DDE is supported for reading data in the DATA step.
- It is very similar to what was shown earlier.
- A worksheet can be read and written to in the same step.
- Allows for very specialized and detailed control from the data step.
- XML can also be produced from Excel.
- SAS can read via an XML libname engine.
- Much more written about XML input.

# What Are Reading Options with SAS/ACCESS?

---



Reading Excel becomes much easier.

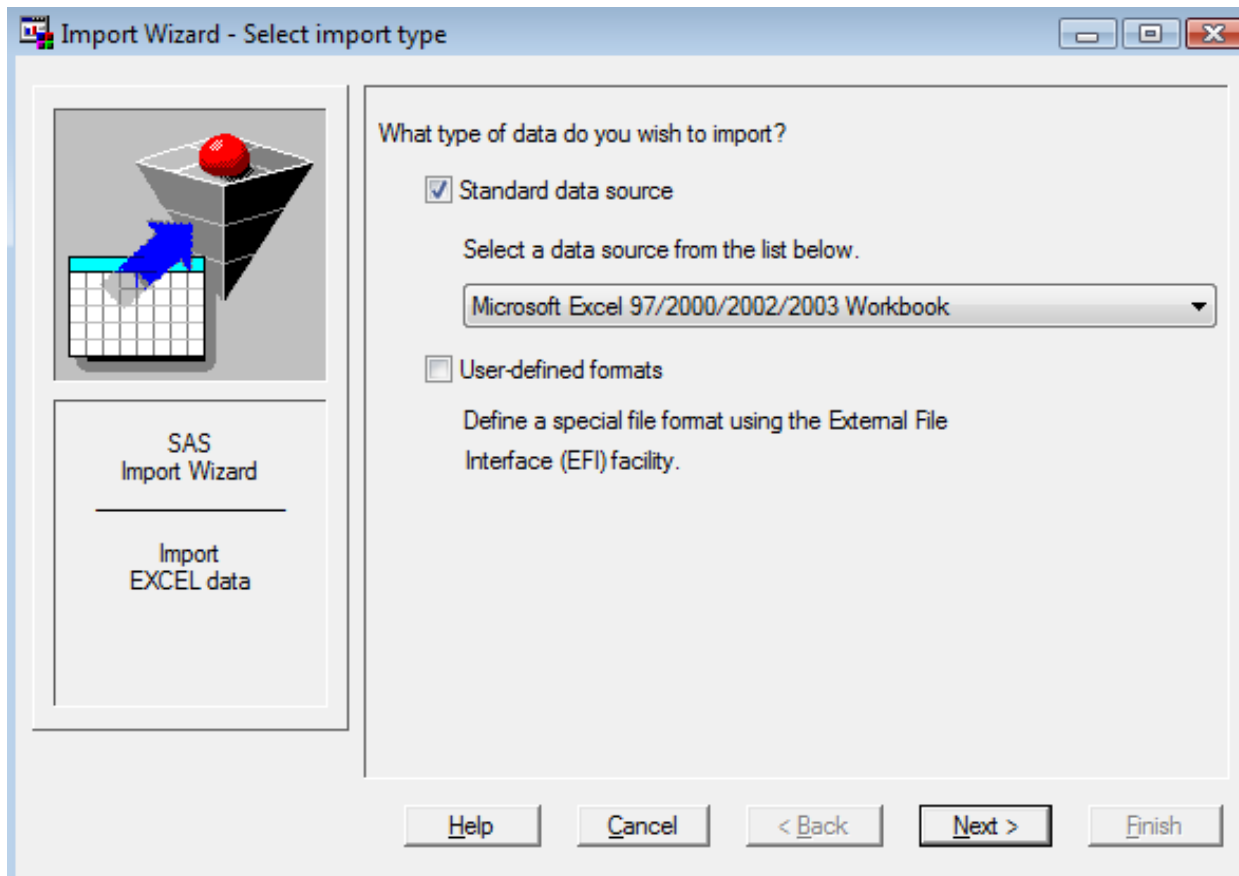
- Import wizards
- PROC IMPORT
- LIBNAME
- Even PROC SQL pass-through to read data.

# How Do PROC IMPORT Wizards Work?



Wizards convert Excel to SAS and are very similar to those shown earlier.

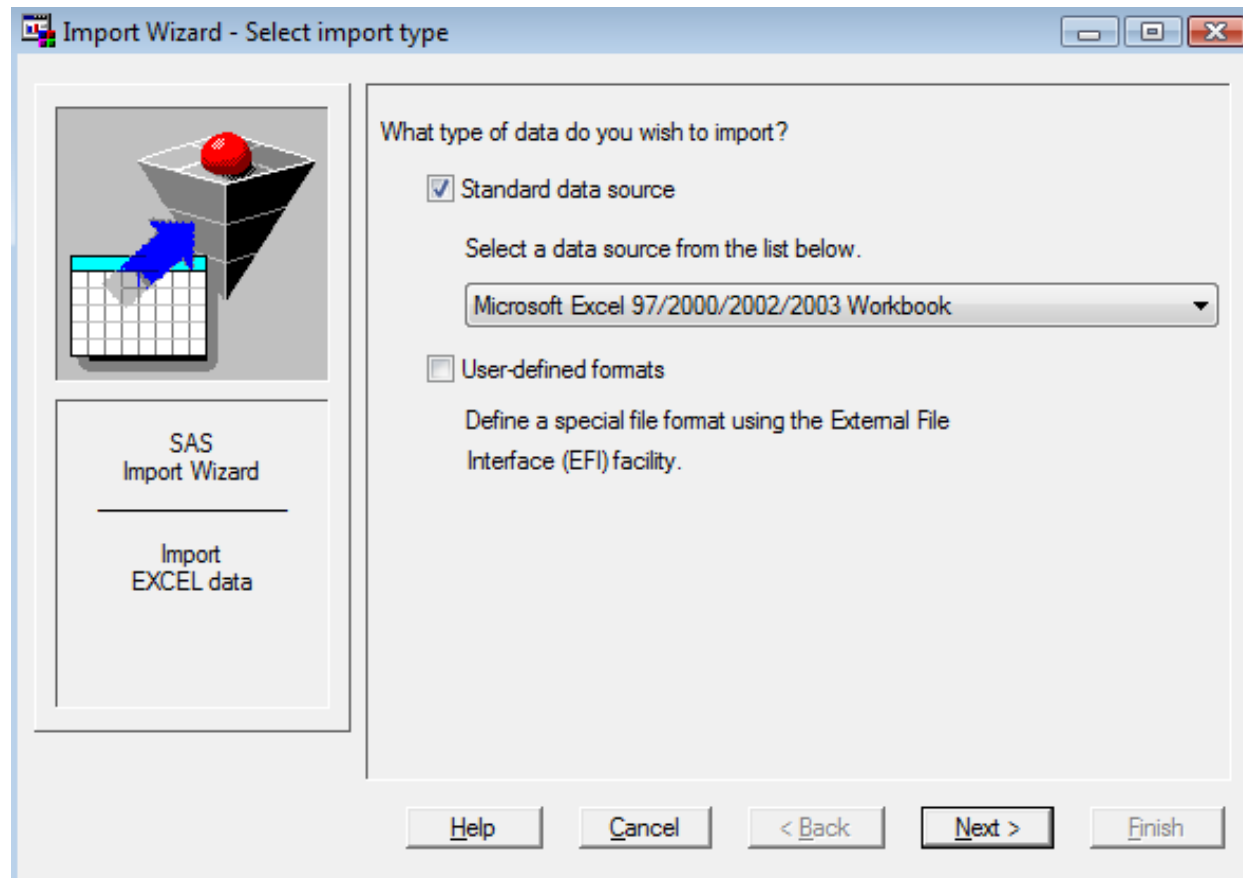
To start the process, use the **File Import Data** menu.



# How Do You Specify Excel as Input?



You can select an import type of Excel and press Next.

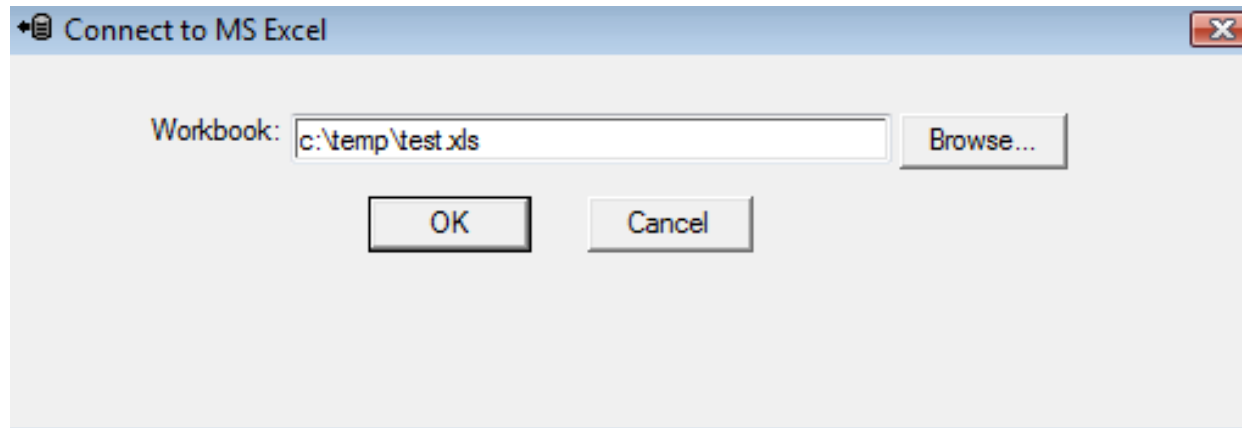


# How Do We Name the Input Workbook?

---



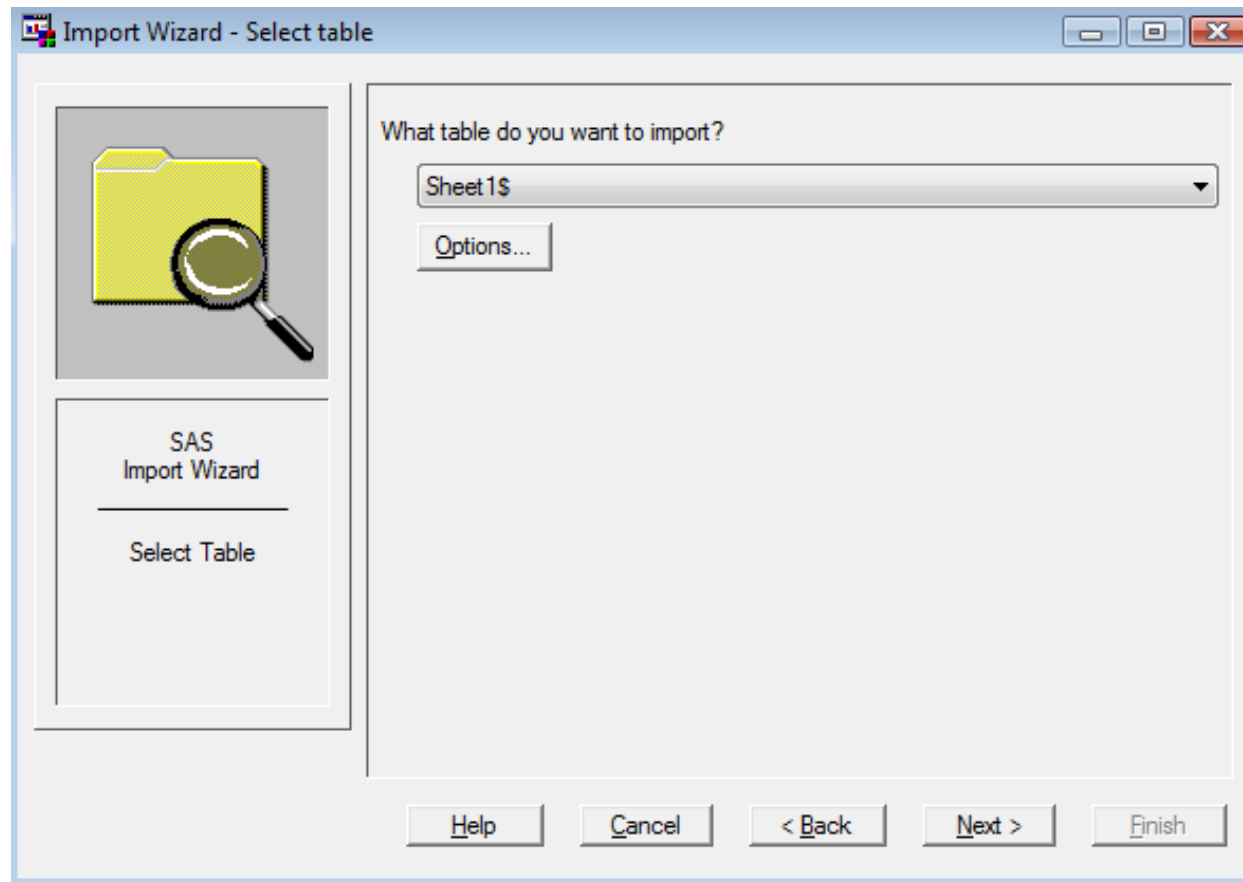
You can then fill in the workbook name, or browse to find it, press OK.



# Can I Specify a Worksheet?



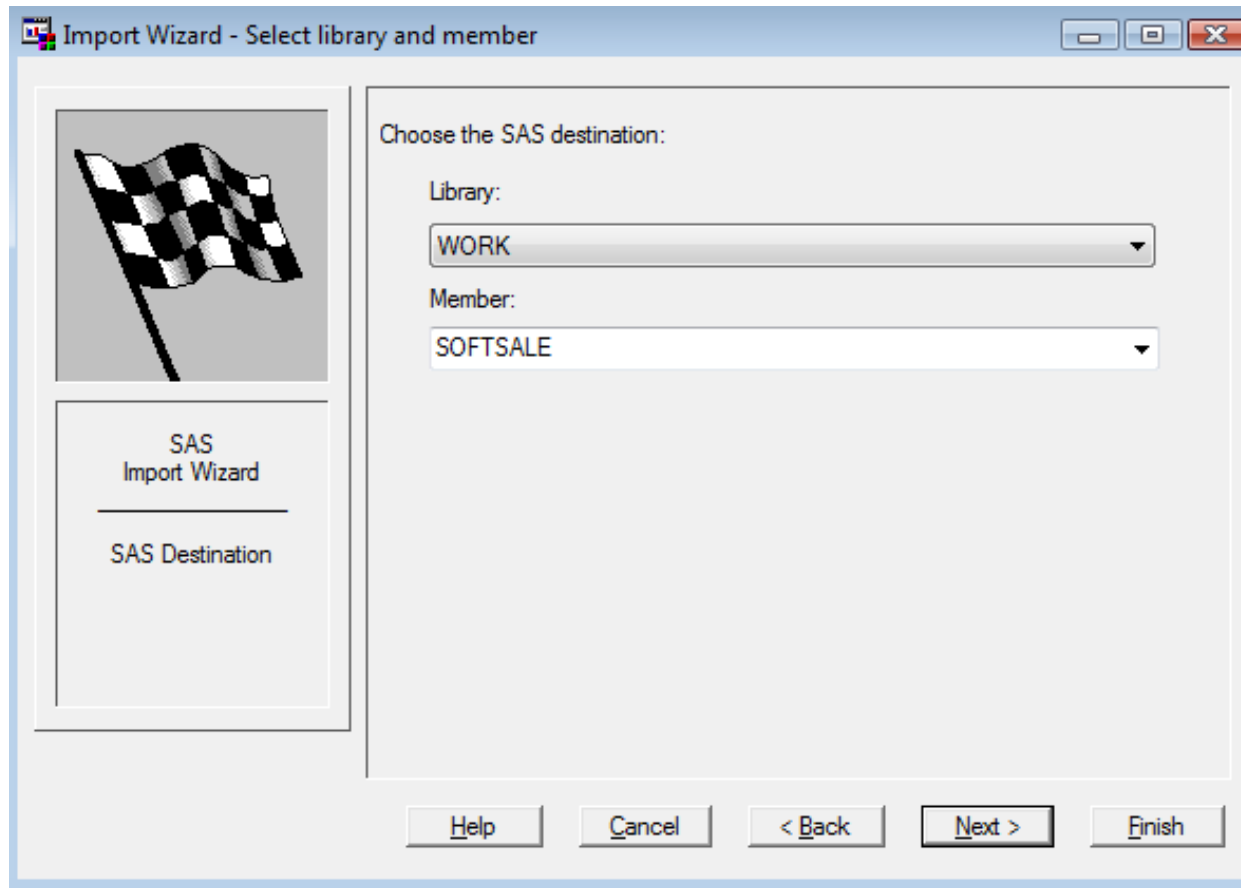
Choose the sheet you would like to import from those shown.



# Where Do We Name Our OUTPUT?



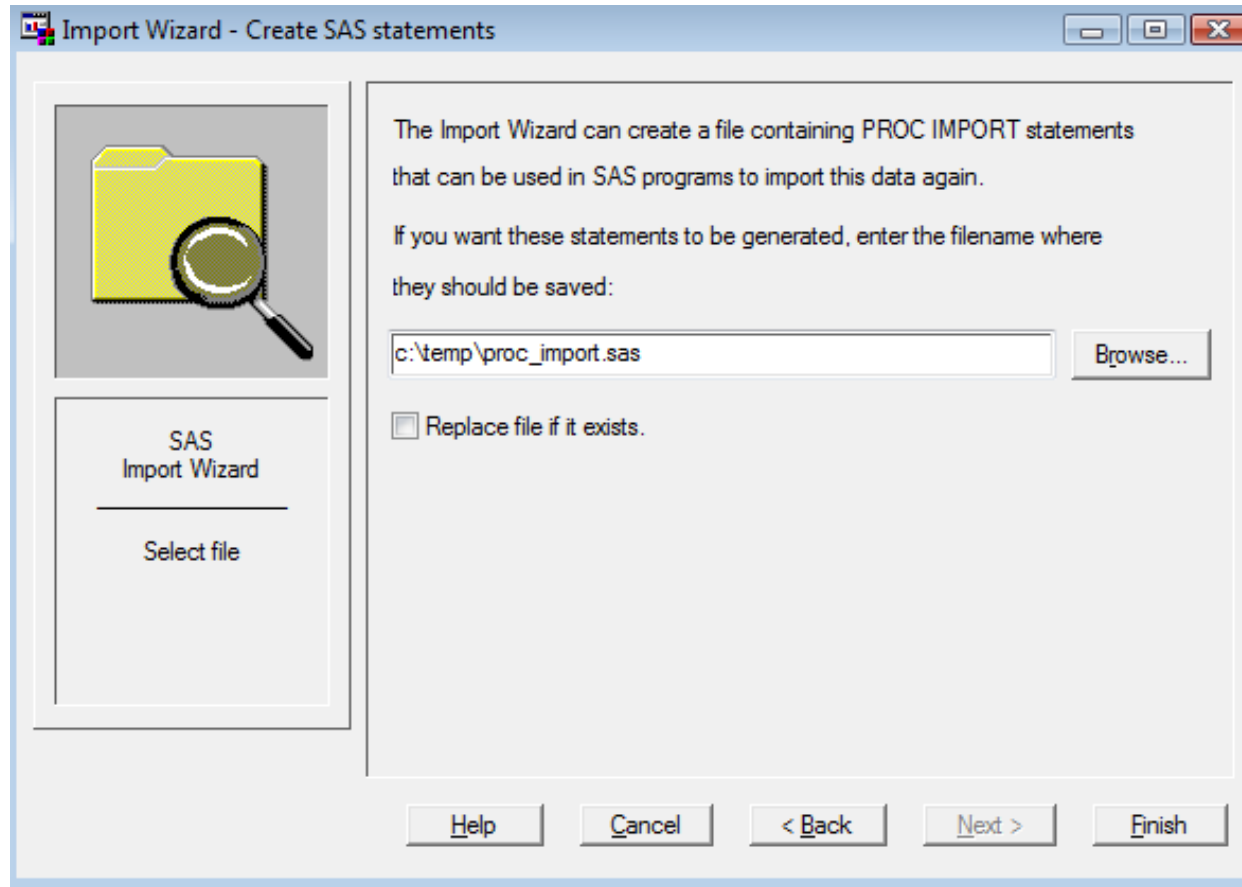
Choose a library and type in a member name.



# Can the Generated Code Be Captured?



You can capture the generated SAS PROC IMPORT for rerun, then Finish.



# What Does the SAS Code Look Like?

---



The PROC IMPORT can be altered and re-run at any time.

```
PROC IMPORT OUT= WORK.SOFTSALE  
    DATAFILE= "c:\temp\test.xls"  
    DBMS=EXCEL REPLACE;  
    RANGE="Sheet1$";  
    GETNAMES=YES;  
    MIXED=NO;  
    SCANTEXT=YES;  
    USEDATE=YES;  
    SCANTIME=YES;  
RUN;
```

# Can We Read Excel Worksheets With Libname?



- LIBNAME makes the workbook appear as a SAS library, and each “named range” appears as a table.
- This is probably the easiest way to access Excel from SAS.

A screenshot of the Microsoft Excel application window. The title bar reads "Microsoft Excel". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains various icons for file operations and calculations. The active cell is G19. The worksheet is named "test.xls" and contains a table with 11 rows and 7 columns (A-G). The data is as follows:

	A	B	C	D	E	F	G
1	Name	Division	Years	Sales	Expense	State	
2	BENJAMIN	S	3	201.11	25.21	IL	
3	JENNIFER	S	1	542.11	134.24	IL	
4	PAT	H	4	4009.21	322.12	IL	
5	ANDREW	S	24	1762.11	476.13	MN	
6	SARAH	S	6	301.21	65.17	MN	
7	TOM	S	5	5669.12	798.15	MN	
8	WILLIAM	H	11	3231.75	644.55	MN	
9	BETH	H	12	4822.12	982.1	WI	
10	CHRIS	H	2	233.11	94.12	WI	
11	JANET	S	1	98.11	125.32	WI	

The status bar at the bottom shows "Ready" and "NUM".

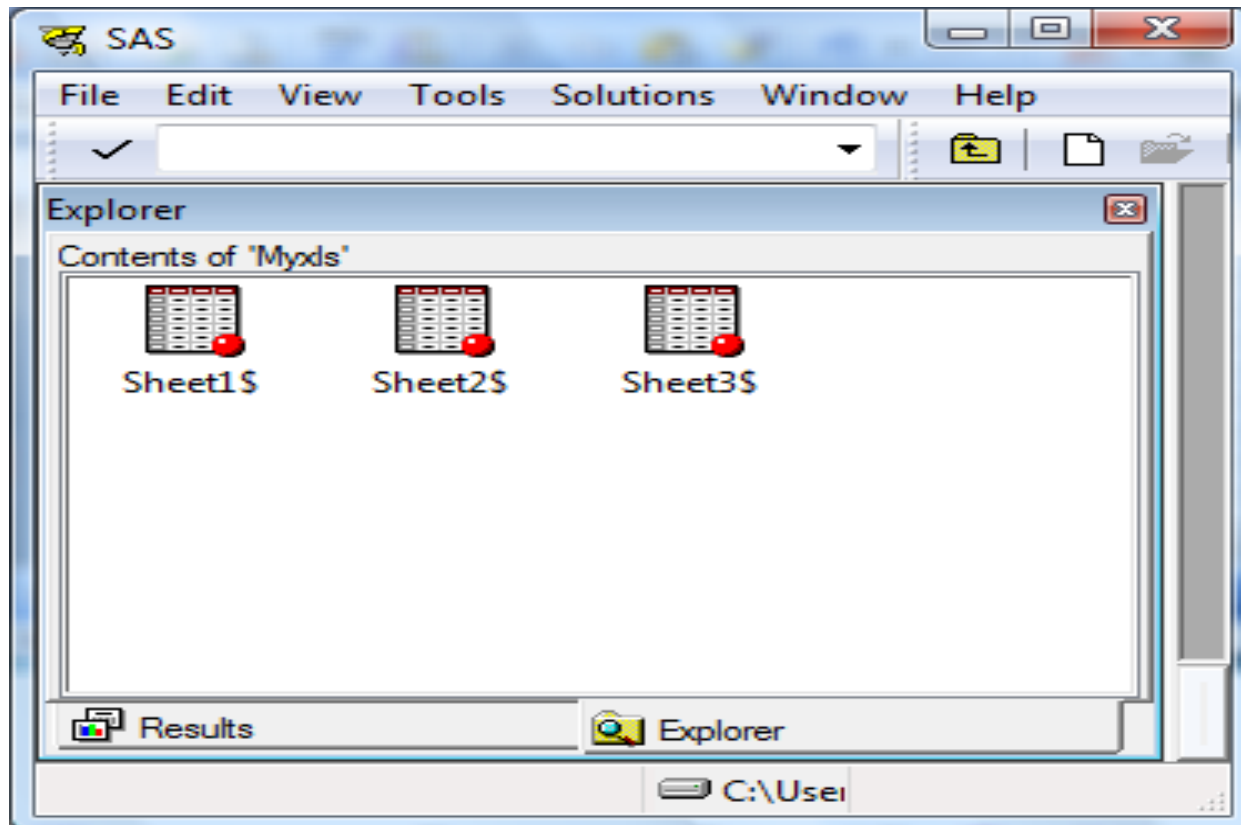
# Do We Need to Set Up Named Ranges?



You can, or use the ones SAS sets up.

- LIBNAME automatically sets up range name (Sheet + \$).
- Can verify that with SAS explorer window or Proc Contents.

```
libname myxls excel path="c:\temp\test.xls";
```



# How Do I Refer To The Table Name In SAS?

---



- Use the name ranges as you would with any SAS dataset.
- Because of the \$ in the name enclose it in quotes, follow with N.
- If you prefer, you could create your own named ranges.

```
proc print data=myxls.'sheet1'$n;  
run;
```

# Can PROC SQL PASS-THROUGH Read Excel?

---



- PROC SQL Pass-Through accesses Excel tables via sub-query.
- Probably easiest when using Excel named ranges.

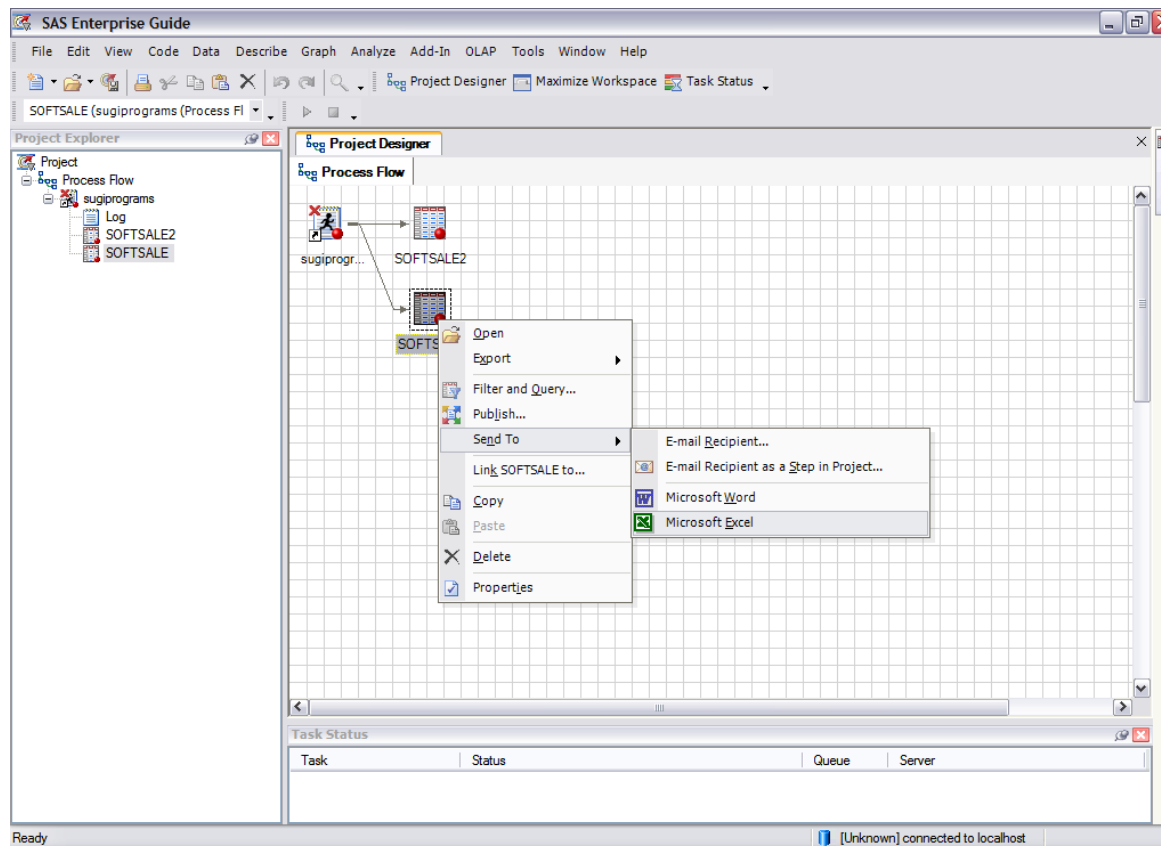
Example: Create a SAS table from a named range “sdata” in our workbook.

```
proc sql dquote=ansi;  
connect to excel (path="c:\temp\test.xls");  
create table work.softsale as  
select * from connection to excel  
    (select * from sdata);  
disconnect from excel;  
quit;
```

# Can I Export Excel Data From Enterprise Guide?



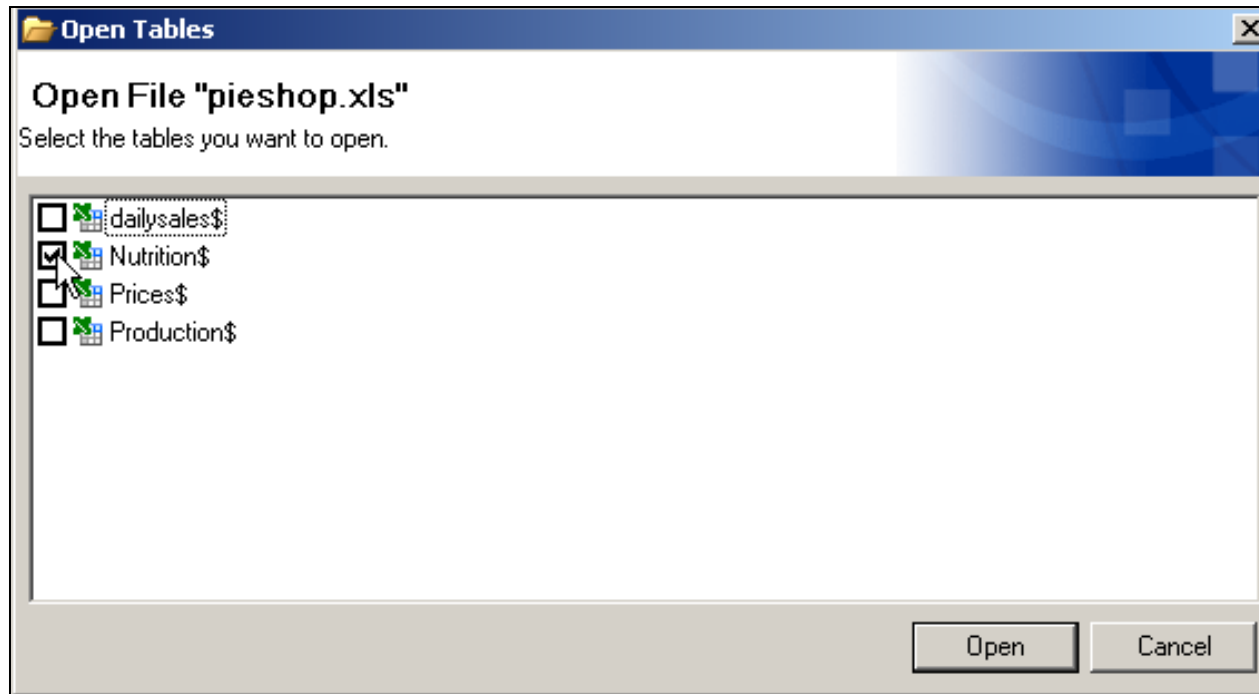
- Exporting data from EG to Excel is very simple!
- Go to your process flow, right click on the dataset you want to export, and select **Send To**, then **Excel**. It will send it to a new workbook.



# Is There An Easy Way to Import Excel Into EG?



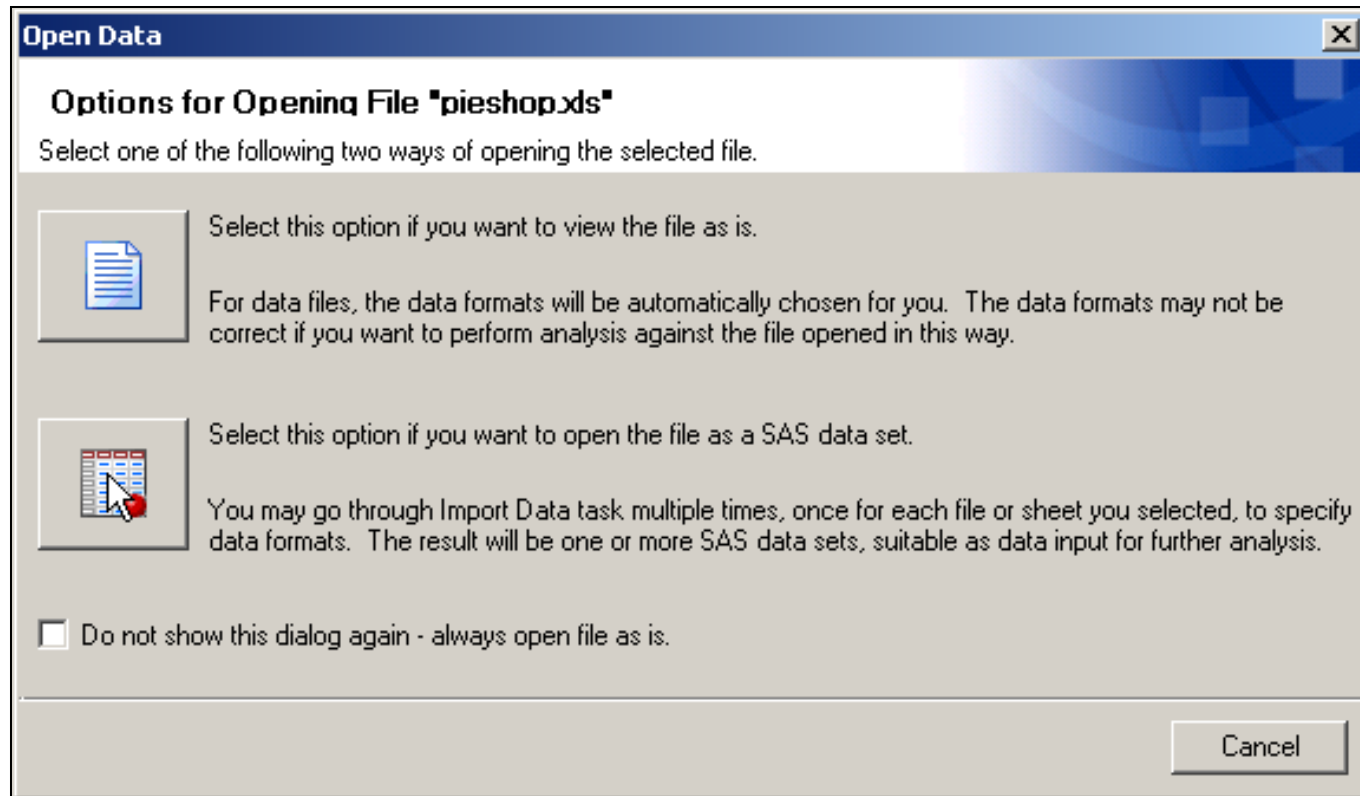
- On the **File** menu, select **Open** → **Data**.
- Select **Local Computer** or **SAS Servers**.
- Navigate to the location of the data set and highlight the data set name.
- Click **Open**
- Check the boxes next to the tables you want to open.
- Click **Open**.



# Can I Control the Import Settings in EG?



- Select **View the file as is** to open the entire table and use default column headings, data types and formats.  
or
- Select **Open the file as a SAS data set** to use the **Import Data Wizard**. This will allow you to control the way the table is imported.

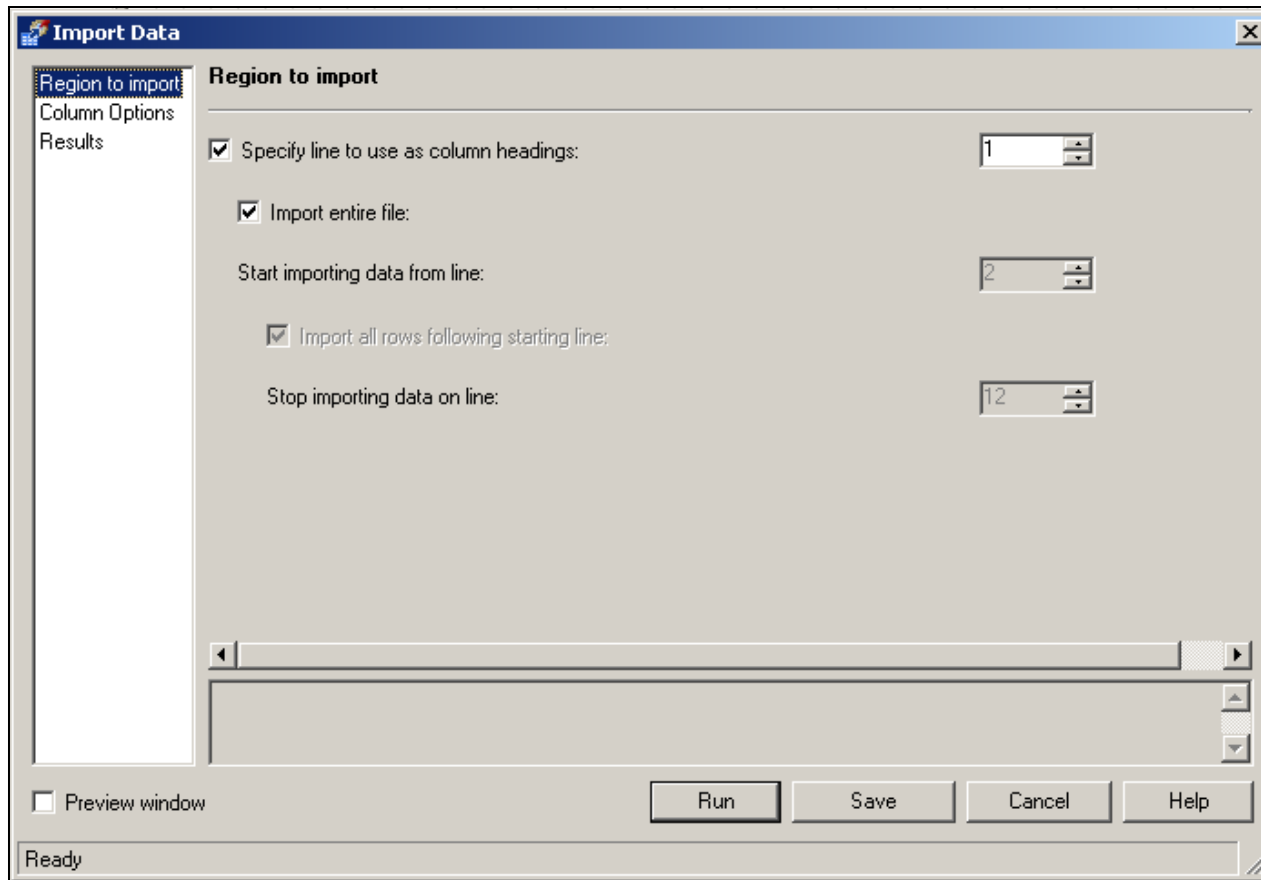


# What Are the Different Options?



**The Import Data Wizard allows you to choose:**

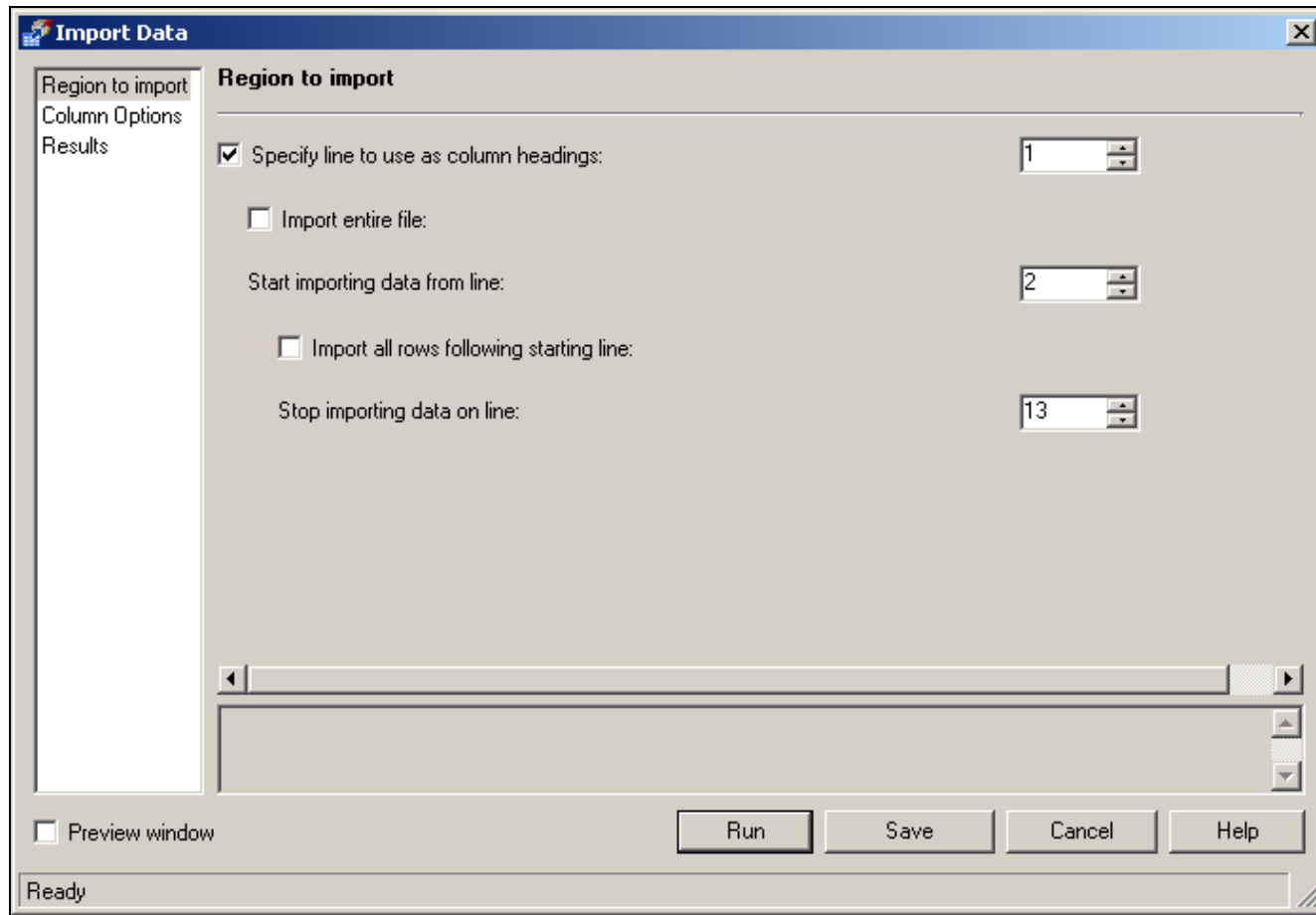
- Rows and columns to import.
- The row to use for column headings.
- Names, types, labels, formats and informat for variables.



# Can I Import a Specific Region From Excel?



- On the **Selection Pane** of the **Import Data Wizard** , highlight **Region to Import**.
- Change **Region to Import** options.



# What Are the Region Options?

---



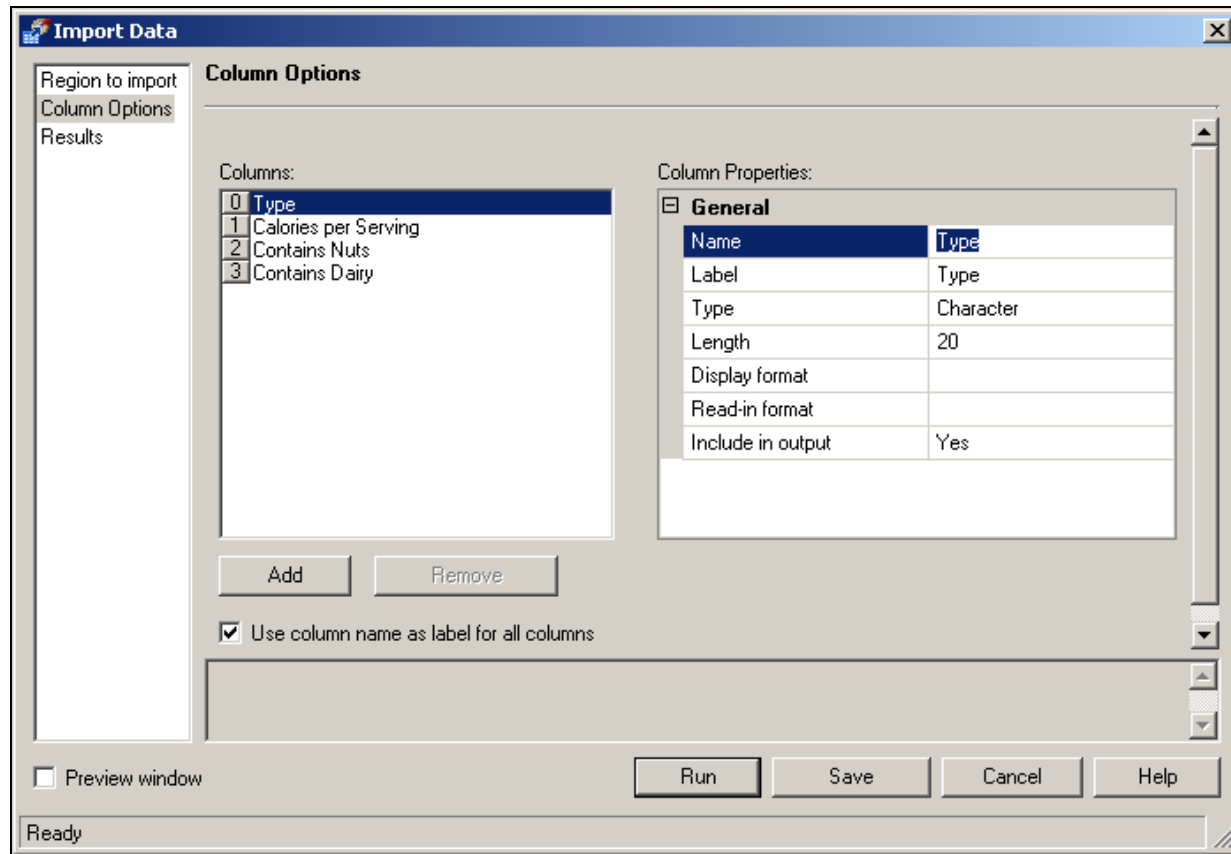
## **These options are available:**

- Choose a line to use as column headers.
- Choose the starting line.
- Choose the ending line.

# What About Setting Column Properties in EG?



- On the **Selection Pane**, highlight **Column Options**.
- On the **Columns** list, highlight the name of a column.
- In the **Column Properties** change the options.



# What Column Properties Can I Change?

---



- **Name:** name of the variable. SAS naming restrictions apply.
- **Label:** text that is sometimes displayed in place of the variable's name.
- **Type:** character or numeric.
- **Length:** maximum length of the data values, in bytes.
- **Informat:** determines how the data is read in.
- **Format:** determines how the data is displayed.
- **Include in output:** determines whether the variable will be included in the resulting SAS data set.

# What Is The SAS Add-In for Microsoft Office?

---



- Part of SAS's comprehensive Enterprise Intelligence Platform.
- Allows business users SAS capabilities within the familiar environment of Microsoft Office.
- Installed as an integrated menu option within Microsoft Excel.

# What Can We Do With the SAS Add-In?

---



## Access Data

- Utilize enterprise data from within MS Office.
- Your SAS data sources.
- Data sources on server (including Oracle, Teradata, and DB2).
- Analyze very large data sources, (over 65,536 Excel 2003 limit) because processing occurs on SAS server.
- Switch dynamically between multiple data sources.

# Can It Do Analysis and Reporting?

---



- Analyze the data and return results back to Excel.
- Programs or “SAS Stored Processes” execute SAS analytics and produce reports.
- Many dialogs for common SAS analytical tasks.
- Refresh data automatically and refresh results with a mouse click.

# Conclusions

---



- SAS provides a variety of tools to both read and write Excel.
- Available products, source data, output requirements, data storage limitations, and programming skills may influence your choices.
- If your current tools don't create what you need, there are many tools available.



## REFERENCES

- “More Tips and Tricks for Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS”, DeIGobbo, SGF 2009
- “De-Mystifying the SAS Libname Engine in Microsoft Excel: A Practical Guide”, Choate and Martell, Sugi 31

## ACKNOWLEDGMENTS

- The authors would like to acknowledge Joy First for editing this paper. We would also like to acknowledge our clients who have exposed us to such a variety of source data and output requests, who have given us the ability to work with many of the different methods of working with SAS and Excel together.

## RECOMMENDED READING

- SAS 9.2 Documentation.



## **SYSTEMS SEMINAR CONSULTANTS, INC.**

SAS® Training, Consulting, & Help Desk Services  
2997 Yarmouth Greenway Drive • Madison, WI 53711  
(608) 278-9964 • Fax (608) 278-0065  
[www.sys-seminar.com](http://www.sys-seminar.com)



---

Steven First

President

[sfirst@sys-seminar.com](mailto:sfirst@sys-seminar.com)

Jennifer First

Director of Operations

[jfirst@sys-seminar.com](mailto:jfirst@sys-seminar.com)