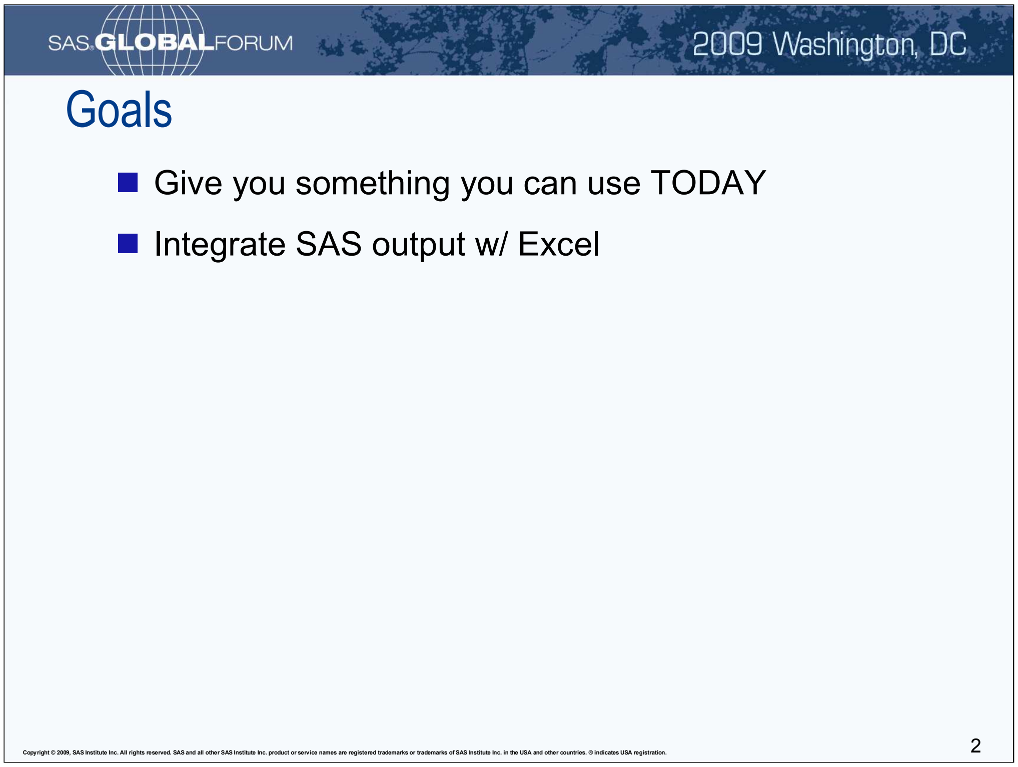
SAS GLOBAL FORUM

More Tips and Tricks for Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS®

Vince DelGobbo
Web Tools Group, SAS Institute Inc.

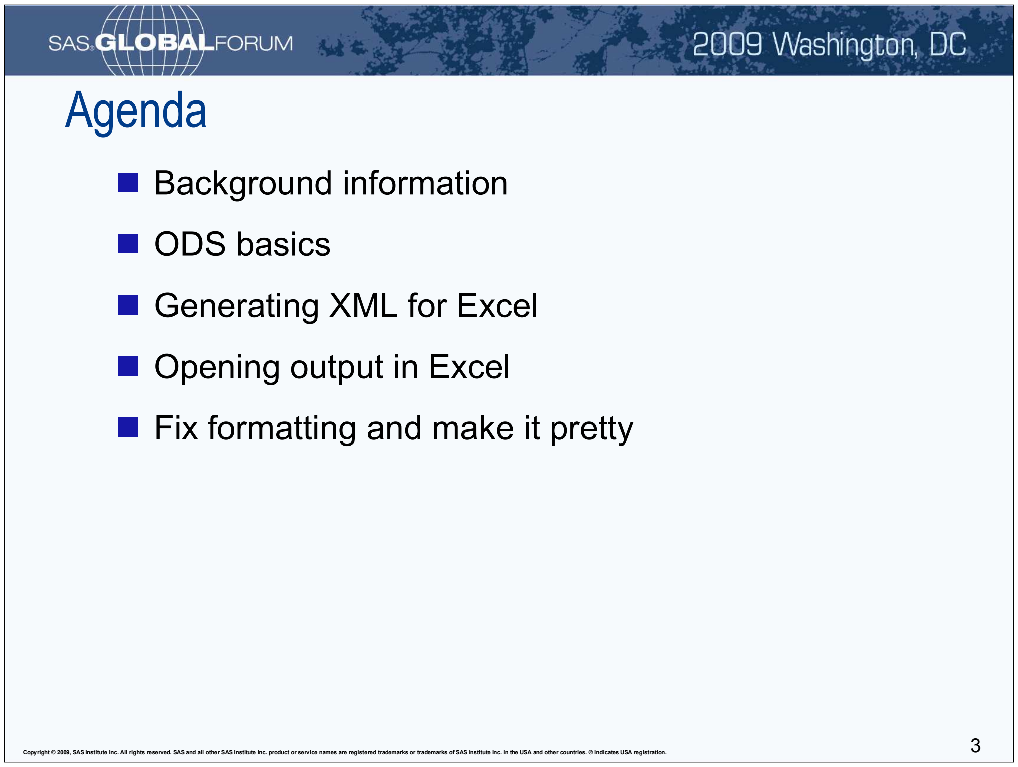
SAS Global Forum 2009
March 22-25, 2009
Gaylord National Resort, National Harbor
Washington, DC

A special *"thank you"* to Peter Eberhardt and Nancy Brucken for inviting me to present this topic, and to Chris Barrett of SAS Institute Inc. for his valuable input on the accompanying paper.



Goals

- Give you something you can use TODAY
- Integrate SAS output w/ Excel

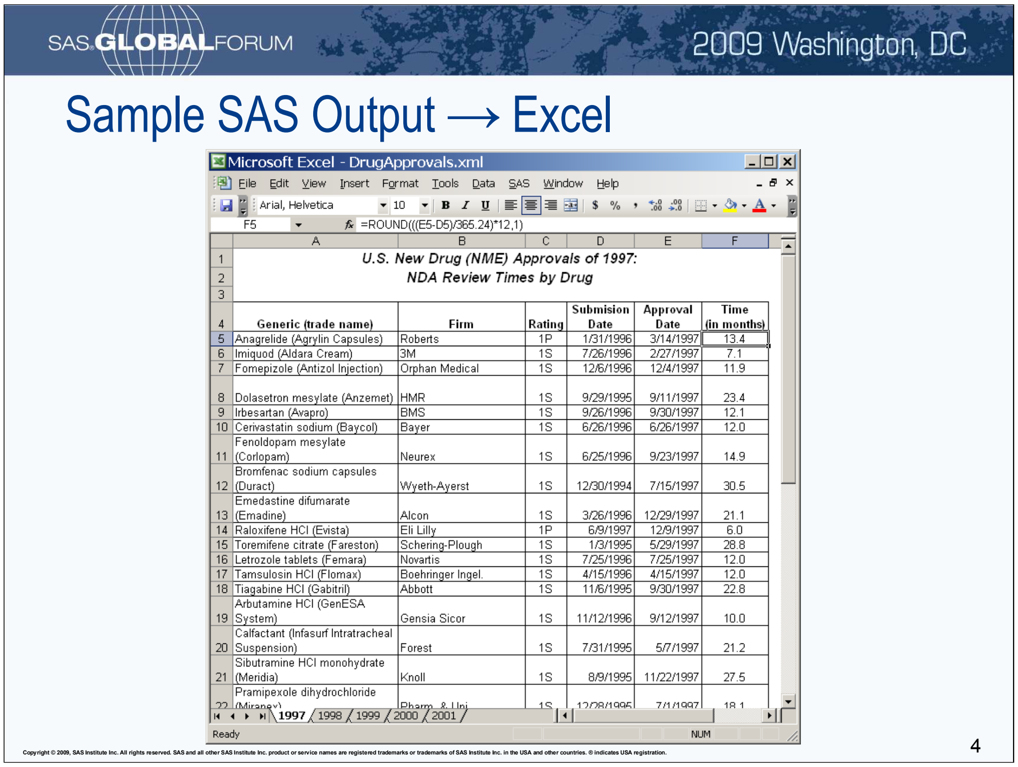


Agenda

- Background information
- ODS basics
- Generating XML for Excel
- Opening output in Excel
- Fix formatting and make it pretty

Software Requirements

- Base SAS, ***any*** operating system.
- SAS 9.1.3 or later.
- **<u>Modified version</u>** of the ExcelXP tagset (see the accompanying paper for details).
- Microsoft Excel XP (a.k.a. Microsoft Excel 2002) or later.



Sample SAS Output ⟶ Excel

Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Arial, Helvetica 10
F5

=ROUND(((E5-D5)/365.24)*12,1)

U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug

	Generic (trade name)	Firm	Rating	Submision Date	Approval Date	Time (in months)
5	Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
6	Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
7	Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
8	Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4
9	Irbesartan (Avapro)	BMS	1S	9/26/1996	9/30/1997	12.1
10	Cerivastatin sodium (Baycol)	Bayer	1S	6/26/1996	6/26/1997	12.0
11	Fenoldopam mesylate (Corlopam)	Neurex	1S	6/25/1996	9/23/1997	14.9
12	Bromfenac sodium capsules (Duract)	Wyeth-Ayerst	1S	12/30/1994	7/15/1997	30.5
13	Emedastine difumarate (Emadine)	Alcon	1S	3/26/1996	12/29/1997	21.1
14	Raloxifene HCl (Evista)	Eli Lilly	1P	6/9/1997	12/9/1997	6.0
15	Toremifene citrate (Fareston)	Schering-Plough	1S	1/3/1995	5/29/1997	28.8
16	Letrozole tablets (Femara)	Novartis	1S	7/25/1996	7/25/1997	12.0
17	Tamsulosin HCl (Flomax)	Boehringer Ingel.	1S	4/15/1996	4/15/1997	12.0
18	Tiagabine HCl (Gabitril)	Abbott	1S	11/6/1995	9/30/1997	22.8
19	Arbutamine HCl (GenESA System)	Gensia Sicor	1S	11/12/1996	9/12/1997	10.0
20	Calfactant (Infasurf Intratracheal Suspension)	Forest	1S	7/31/1995	5/7/1997	21.2
21	Sibutramine HCl monohydrate (Meridia)	Knoll	1S	8/9/1995	11/22/1997	27.5
22	Pramipexole dihydrochloride (Mirapex)	Pharm. & Upj	1S	12/28/1995	7/1/1997	18.1

This is the SAS output that has been incorporated into Excel. The workbook was created using the PRINT procedure, and has 5 worksheets containing drug approval data for the years 1997 through 2001 (Mathieu, 2002).

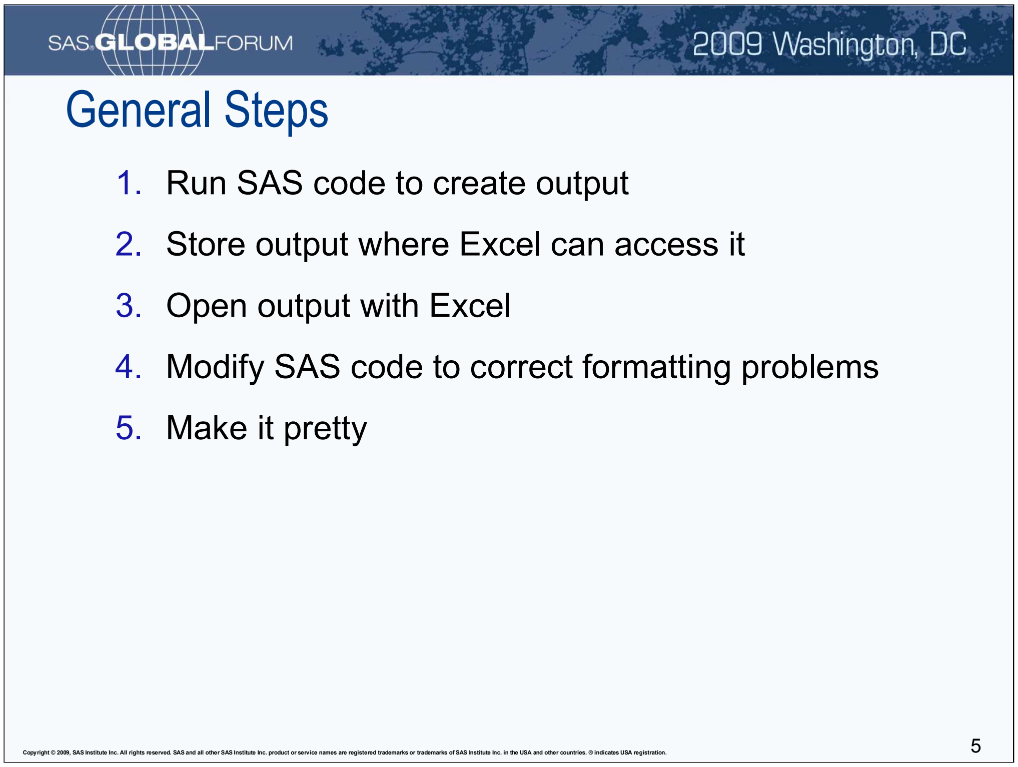
Excel formats, not SAS formats, control the appearance of the numeric and date values. The date values are SAS date values that have been converted to Excel date values, without modifying the underlying SAS data.

Note that the values in the column labeled "Time (in months)" in are not static values, but are calculated from the date values using an Excel function. For example, the value in cell F5 is not 13.4, but instead is the Excel formula `=ROUND(((E5-D5)/365.24)*12,1)`.

This output was generated completely by ODS – there was no "hand-editing" of the Excel workbook. Furthermore, SAS can generate this type of output regardless of the platform – Windows, UNIX, OpenVMS, or z/OS.

Reference:

Mathieu, Mark P., ed. 2002. *PAREXEL's Pharmaceutical R&D Statistical Sourcebook, 2002/2003*. Waltham, MA: PAREXEL International Corporation.

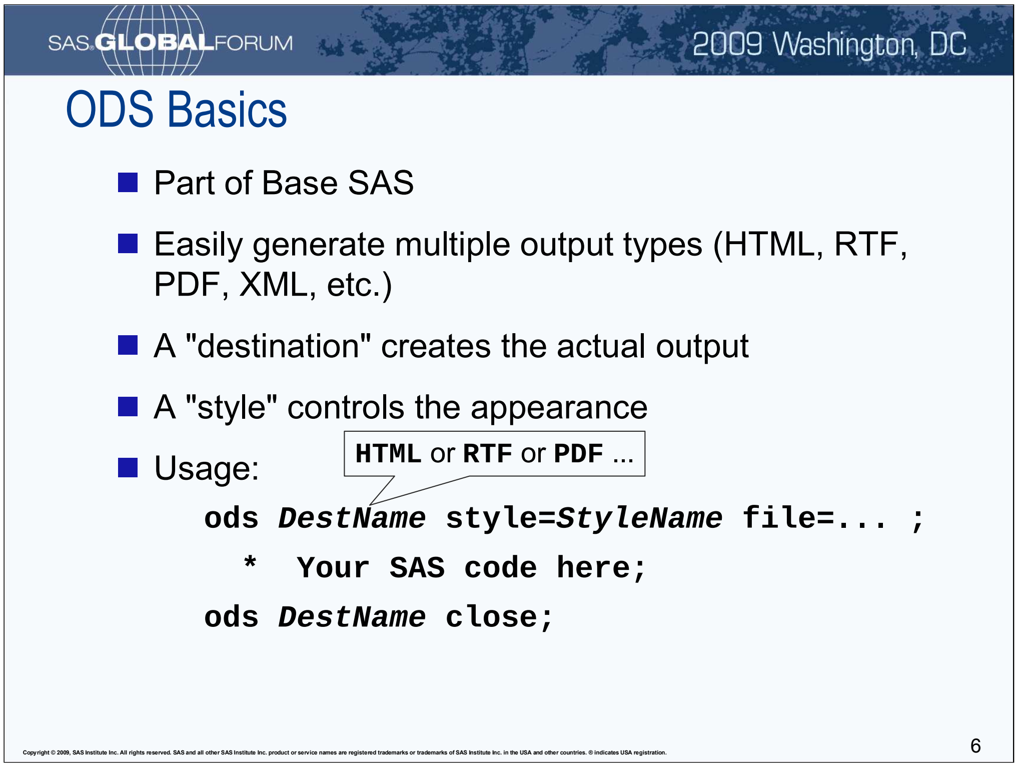


General Steps

1. Run SAS code to create output
2. Store output where Excel can access it
3. Open output with Excel
4. Modify SAS code to correct formatting problems
5. Make it pretty

We use ODS to create an XML file that is stored in a location where Excel can access it. In your production system, SAS and Excel may reside on two different machines. Thus, you may have to make use of network drives, FTP or some other means to move the SAS output to a location so that Excel can access it.

Then the ODS output is opened using Excel. If you have ever done this before, you have probably encountered formatting problems. We fix those problems, and then explore techniques to instruct ODS to create output that Excel is happy with.


SAS GLOBAL FORUM
2009 Washington, DC

ODS Basics

- Part of Base SAS
- Easily generate multiple output types (HTML, RTF, PDF, XML, etc.)
- A "destination" creates the actual output
- A "style" controls the appearance
- Usage:

HTML or **RTF** or **PDF** ...

```
ods DestName style=StyleName file=... ;
  *  Your SAS code here;
ods DestName close;
```

An ODS *destination* creates the actual output (HTML, RTF, PDF, XML, etc.) while an ODS *style* controls the appearance (colors, fonts, border lines, etc.).

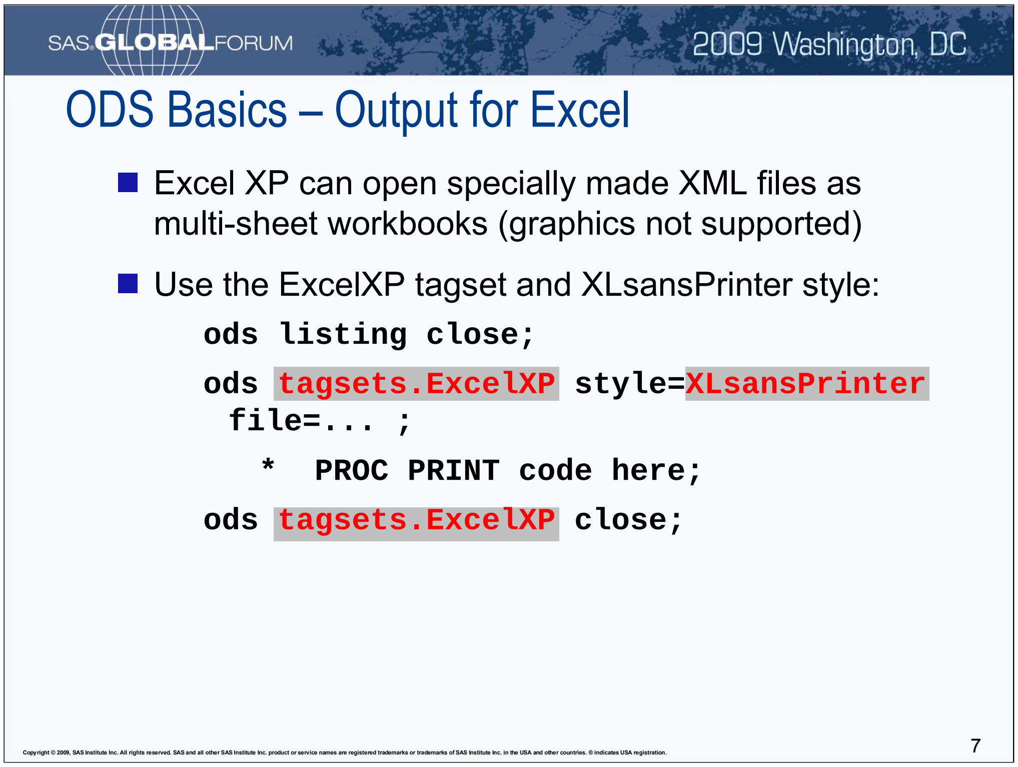
Both a destination and a style are needed to generate output. If you do not specify a style, the style named "Default" will be used.

We use a special type of ODS destination called a "tagset". ODS tagsets can be modified to meet your specific needs using the TEMPLATE procedure. And you can use the TEMPLATE procedure to create your own tagsets.

Reference:

SAS Institute Inc. "The TEMPLATE Procedure: Overview".
http://support.sas.com/onlinedoc/913/docMainpage.jsp?_topic=odsug.hlp/a001020001.htm

(Ignore wrapping in URL above)



ODS Basics – Output for Excel

- Excel XP can open specially made XML files as multi-sheet workbooks (graphics not supported)
- Use the ExcelXP tagset and XLsansPrinter style:

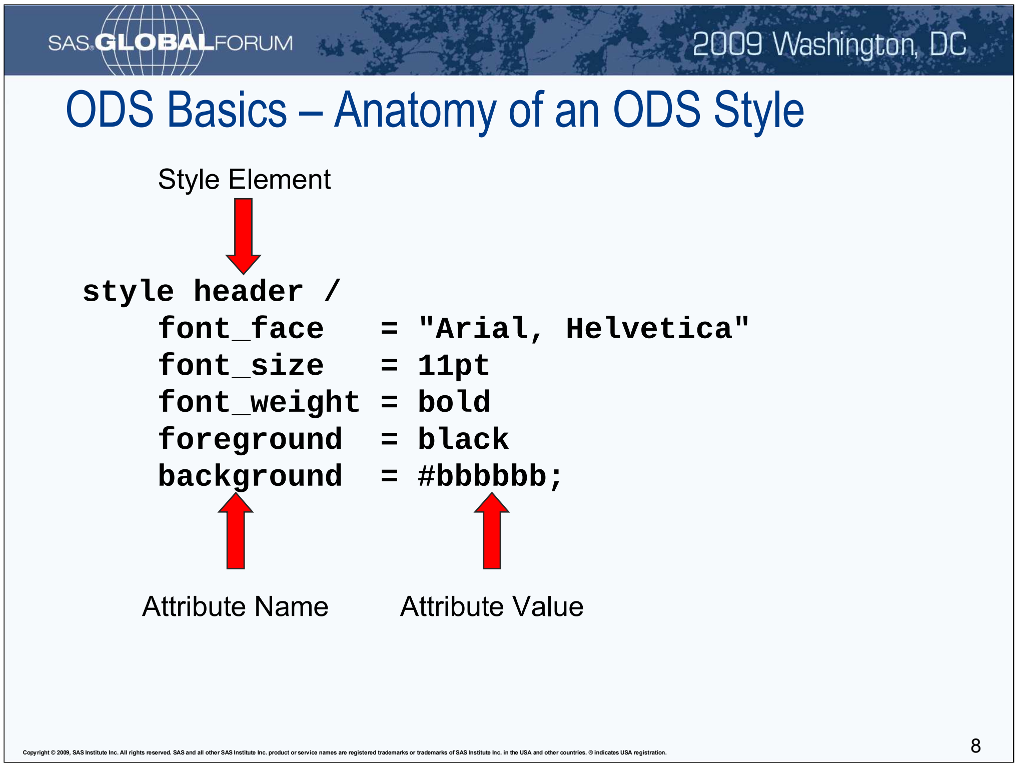
```
ods listing close;

ods tagsets.ExcelXP style=XLsansPrinter
  file=... ;

    *  PROC PRINT code here;

ods tagsets.ExcelXP close;
```

Currently, the Microsoft XML Spreadsheet Specification does not support graphics. Thus, SAS/GRAPH procedure output is not supported. This is a Microsoft Excel limitation, not a limitation of the ExcelXP tagset.

We use the ODS tagset named ExcelXP to create XML output that can be opened using Microsoft Excel XP and later. When opened with Excel, the XML file will be rendered as a multi-sheet Excel workbook. Additionally, we will use an ODS style named XLsansPrinter to provide a look similar to the standard sansPrinter style that is shipped with Base SAS software.

Reference:

Microsoft Corporation. "XML Spreadsheet Reference". http://msdn2.microsoft.com/en-us/library/aa140066(office.10).aspx.



SAS GLOBAL FORUM 2009 Washington, DC

ODS Basics – Anatomy of an ODS Style

Style Element

```
style header /
    font_face   = "Arial, Helvetica"
    font_size   = 11pt
    font_weight = bold
    foreground  = black
    background  = #bbbbbb;
```

Attribute Name

Attribute Value

A style is composed of many *style elements,* each of which control the appearance of a particular part of the output. For example, all styles contain a style element named "header" that controls the appearance of column headings.

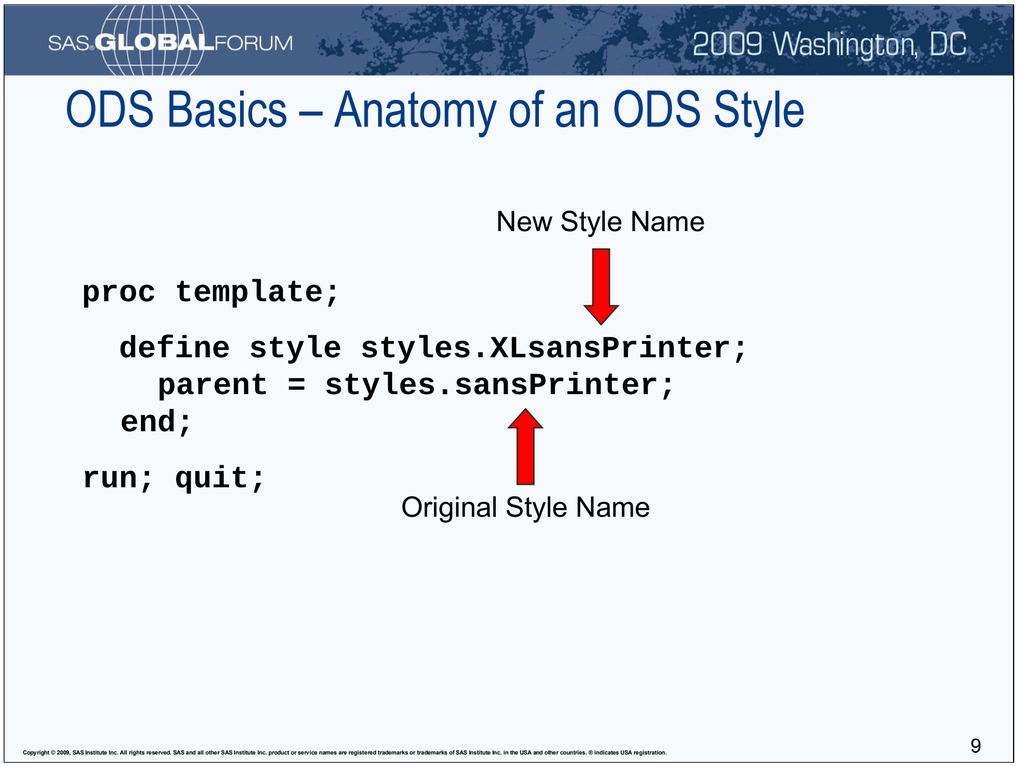
Style elements consist of collections of *style attributes*, such as the background color and font size. The definition of a style element named "header" might look like what is show here.

You can use the TEMPLATE procedure, which is part of Base SAS, to change the attributes of style elements, create new style elements or even create completely new styles.

Reference:

SAS Institute Inc. "The TEMPLATE Procedure: Creating a Style Definition". http://support.sas.com/onlinedoc/913/docMainpage.jsp?_topic=odsug.hlp/a002565239.htm

(Ignore wrapping in URL above)



SAS GLOBAL FORUM 2009 Washington, DC

ODS Basics – Anatomy of an ODS Style

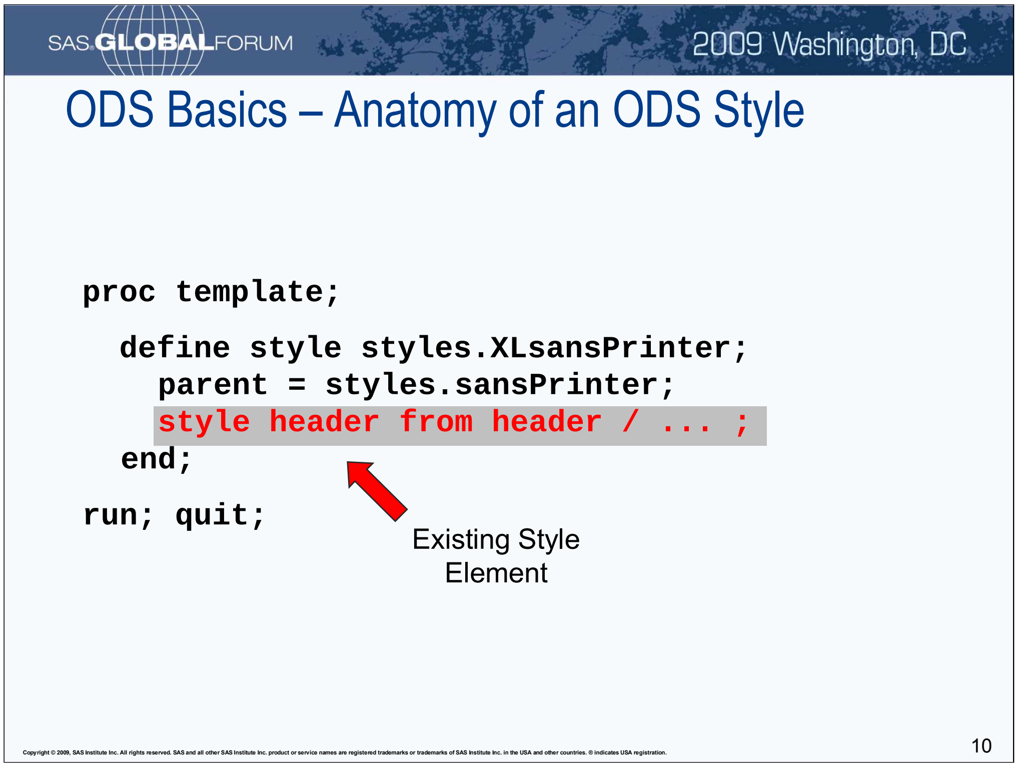
New Style Name

```
proc template;
  define style styles.XLsansPrinter;
    parent = styles.sansPrinter;
  end;
run; quit;
```

Original Style Name

This is a partial listing of the TEMPLATE procedure code used to create the XLsansPrinter style. As you can see, it is based on the sansPrinter style that is shipped with Base SAS.

At this point, the XLsansPrinter style is identical to the sansPrinter style. The XLsansPrinter style contains all the same style elements and style attributes as the sansPrinter style.

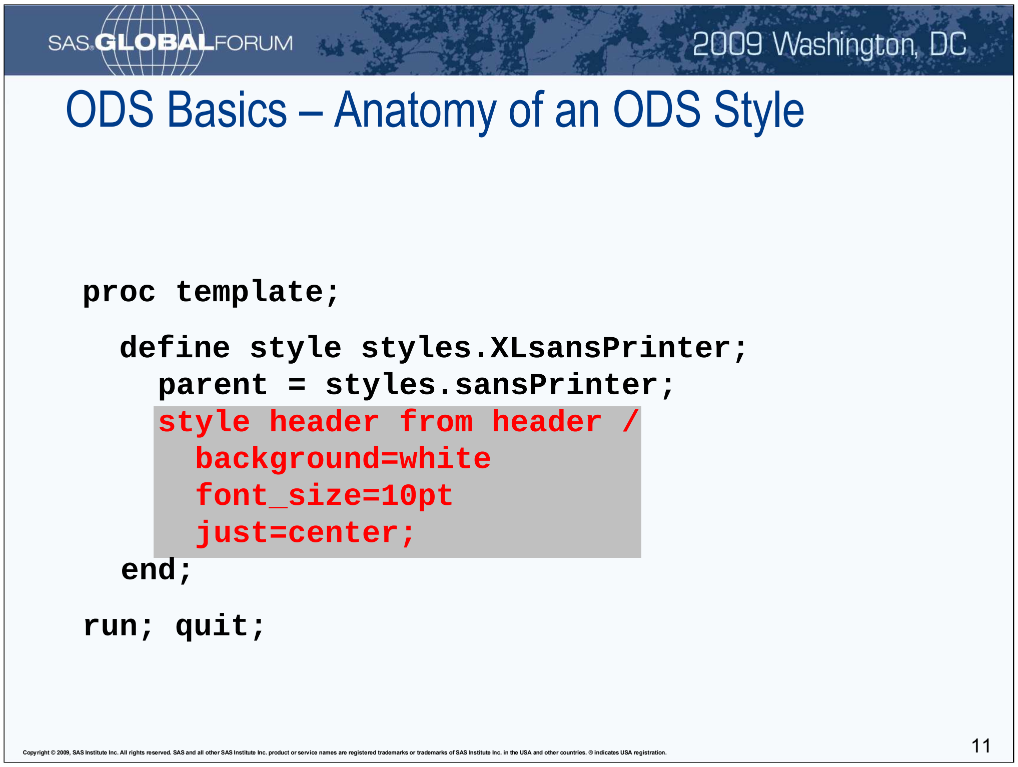


ODS Basics – Anatomy of an ODS Style

```
proc template;
  define style styles.XLsansPrinter;
    parent = styles.sansPrinter;
    style header from header / ... ;
  end;
run; quit;
```

Existing Style Element

Here is an example of how you would modify the standard style element named "header". This style element is used to control the appearance of column headings in procedure output. Thus, any time the XLsansPrinter style is used, the appearance of the column headings will be controlled by the attributes specified in this style element.

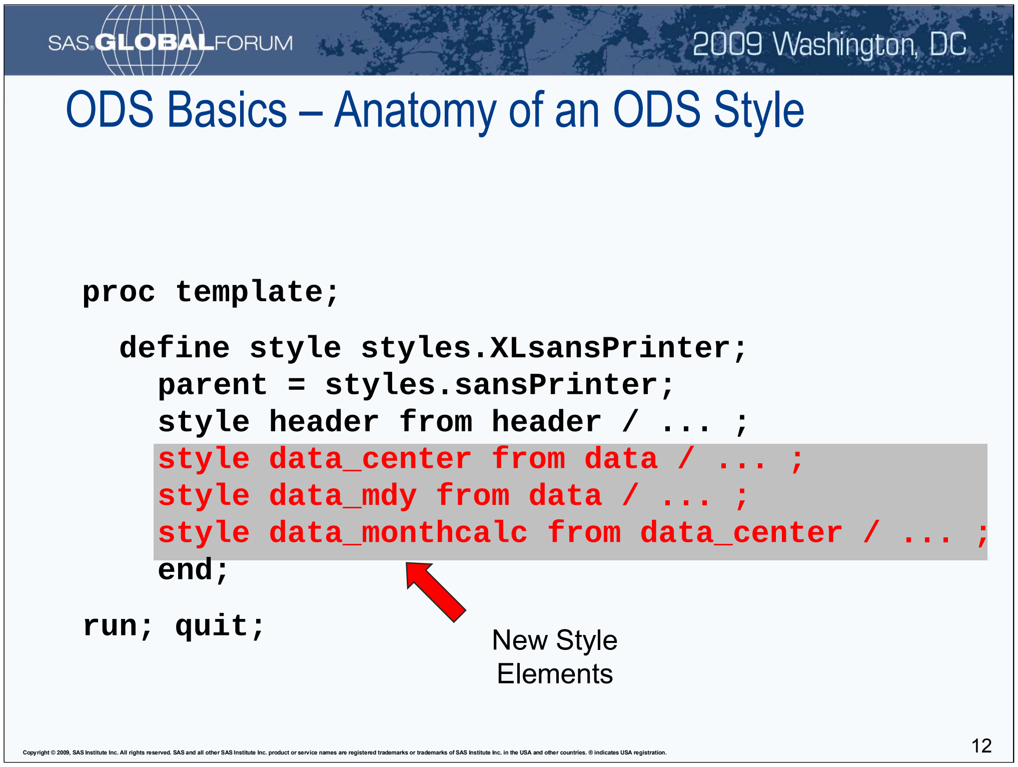


SAS GLOBAL FORUM 2009 Washington, DC

ODS Basics – Anatomy of an ODS Style

```
proc template;
  define style styles.XLsansPrinter;
    parent = styles.sansPrinter;
    style header from header /
      background=white
      font_size=10pt
      just=center;
  end;
run; quit;
```

This code causes the background color to be white, and text will appear centered, using a 10-point font. All other attributes of this style element, such as the foreground color, are inherited from the original "header" style element, and remain unchanged.



SAS GLOBAL FORUM 2009 Washington, DC

ODS Basics – Anatomy of an ODS Style

```
proc template;
  define style styles.XLsansPrinter;
    parent = styles.sansPrinter;
    style header from header / ... ;
    style data_center from data / ... ;
    style data_mdy from data / ... ;
    style data_monthcalc from data_center / ... ;
    end;
run; quit;
```

New Style
Elements

The XLsansPrinter style now contains 3 new style elements:

- "data_center" – based on the "data" style element
- "data_mdy" – based on the "data" style element
- "data_monthcalc" – based on the new "data_center" style element

Further information about the attributes associated with these style elements will be presented later.

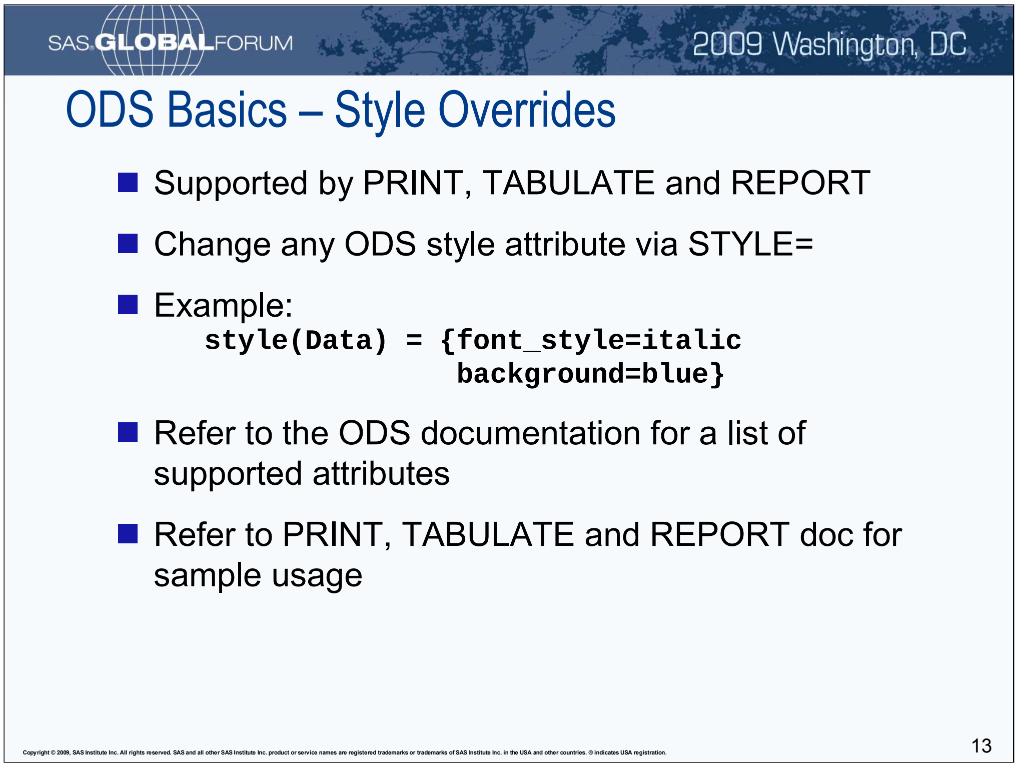
Because of their unique names, ODS does not use these style elements until they are specified by you via a style override.

We will now move on to the topic of style overrides.

Reference:

SAS Institute Inc. "TEMPLATE Procedure: Creating a Style Definition". http://support.sas.com/onlinedoc/913/docMainpage.jsp?_topic=odsug.hlp/a002565239.htm

(Ignore wrapping in URL above)



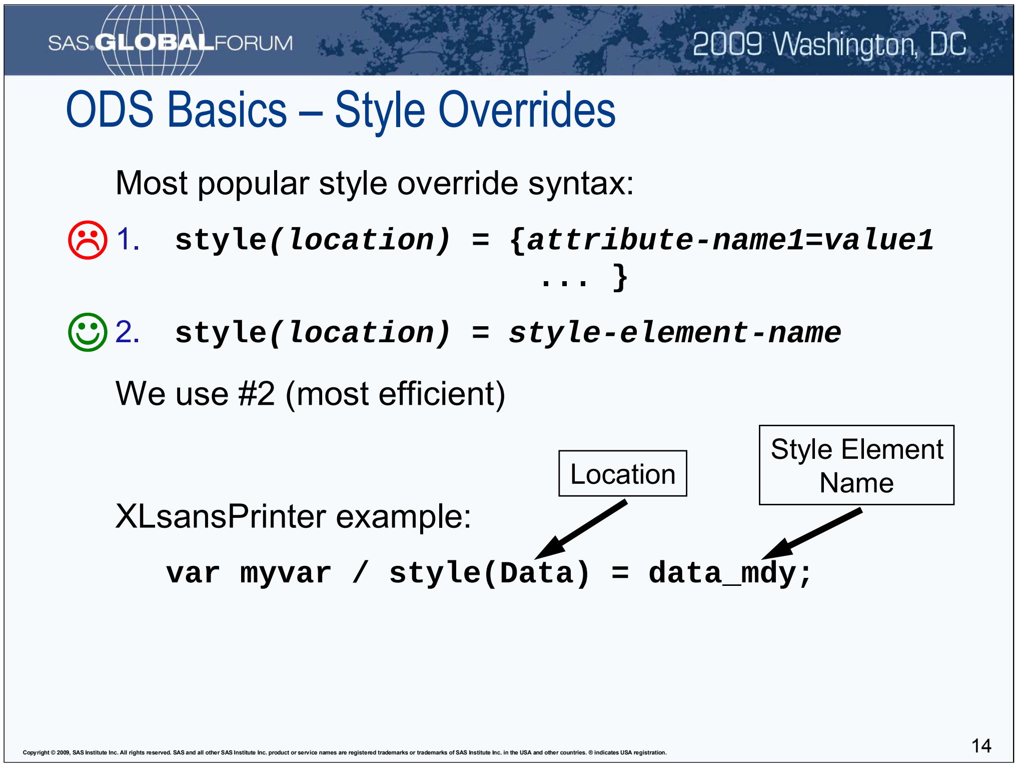
ODS Basics – Style Overrides

- Supported by PRINT, TABULATE and REPORT
- Change any ODS style attribute via STYLE=
- Example:

```
style(Data) = {font_style=italic
               background=blue}
```

- Refer to the ODS documentation for a list of supported attributes
- Refer to PRINT, TABULATE and REPORT doc for sample usage

ODS style overrides are used to override attributes of the ODS style you are using. They are intended to be used to make small changes to the appearance of your output, and should be used sparingly. You could use a style override to change the fonts or colors used by the XLsansPrinter style for the "Data" location of PROC PRINT, as shown here. But there is a better way.



ODS Basics – Style Overrides

Most popular style override syntax:

☹ 1. `style(location) = {attribute-name1=value1 ... }`

☺ 2. `style(location) = style-element-name`

We use #2 (most efficient)

XLsansPrinter example:

Location

Style Element Name

```
var myvar / style(Data) = data_mdy;
```

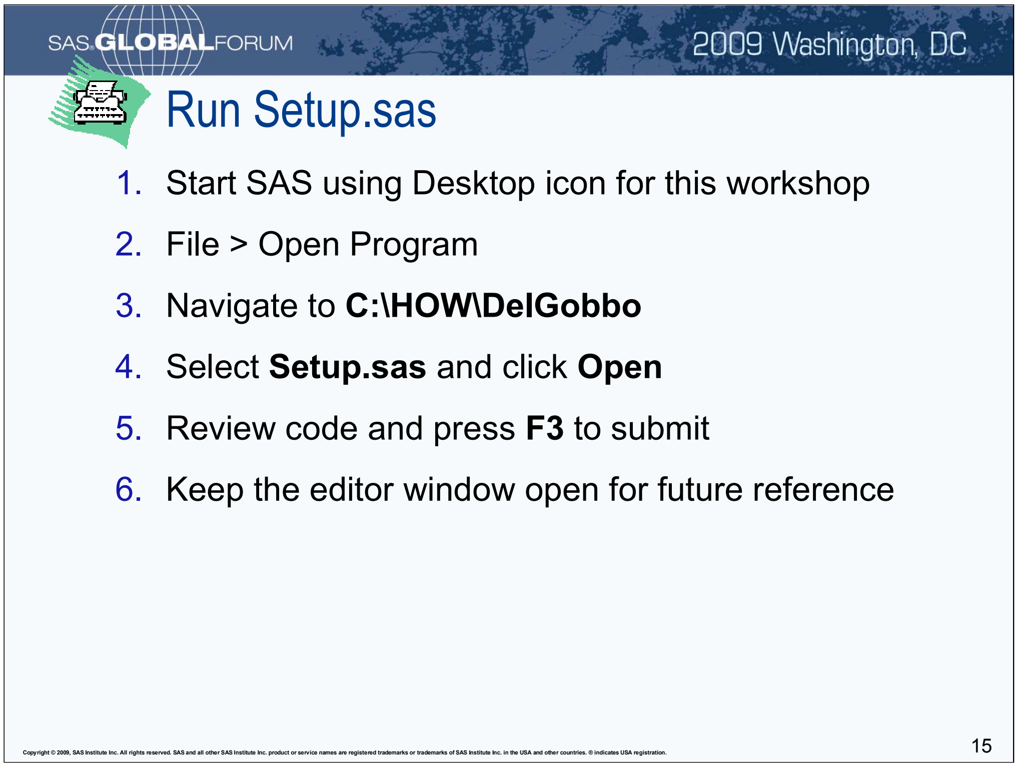
This is the syntax for using style overrides.

The first form is what you might use to change the fonts or colors used by a style element, as shown on the previous slide. While this form is the one that is most commonly used, it is also the least efficient. That is because, if overused, it can cause ODS to create files that are much larger than they would be if the style override was not used.

The second form is the most efficient, and is the one that we use. It is used to associate a pre-defined ODS style element with a particular part of the output. You create new style elements with the desired attributes, as shown earlier, then reference the style element names. This is what we do using the XLsansPrinter style.

The XLsansPrinter style defined in the file "Setup.sas" contains a number of different style elements. Later, we associate these style elements with discrete parts of the SAS output ("locations").

Locations will be discussed later.



Run Setup.sas

1. Start SAS using Desktop icon for this workshop
2. File > Open Program
3. Navigate to **C:\HOW\DelGobbo**
4. Select **Setup.sas** and click **Open**
5. Review code and press **F3** to submit
6. Keep the editor window open for future reference

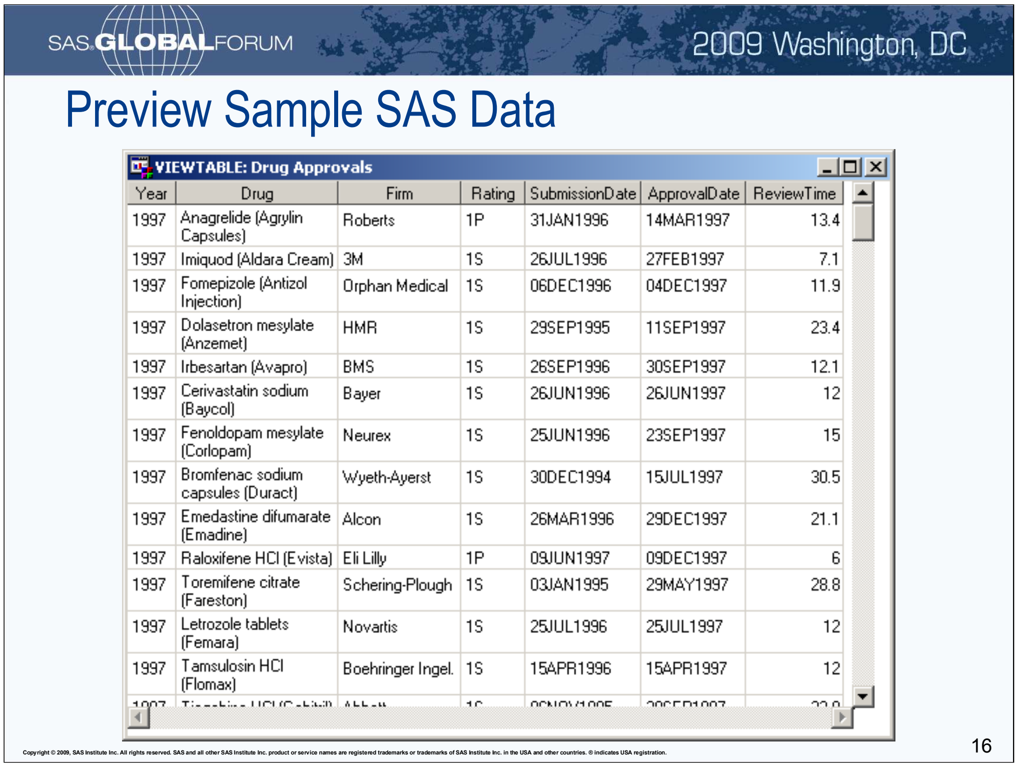
The SAMPDIR global macro variable specifies the directory containing our sample code and data, as well as the ODS-generated XML output.

The program assigns a SAS library (SAMPLE) for the input data and one (MYLIB) for the output ODS tagsets and styles that we create. While it is OK to use the WORK or SASUSER libraries to temporarily store your tagsets and styles, it is a good idea use a different, permanent library that is publicly accessible if you want to make them available to others, or want to reuse them in future programs.

The ODS PATH statement specifies the locations of, and the order in which to search for, ODS tagsets and styles. Because ODS searches the path in the order given, and the access mode for "mylib.tmplmst" is update, PROC TEMPLATE will create and store tagsets and styles in the directory that is associated with the MYLIB library.

The ODS "ExcelXP" tagset has undergone many revisions since SAS 9.1 was released, so we will import a newer copy and store it in the MYLIB library. **Be sure to use a recent version of the ExcelXP tagset in your production code.** (See the accompanying paper for details.)

Finally, the XLsansPrinter style, which produces output similar in appearance to the standard sansPrinter style, is created. Note that the standard "header" style element is overridden, and new style elements "data_center", "data_mdy" and "data_monthcalc" are created.



Preview Sample SAS Data

Year	Drug	Firm	Rating	SubmissionDate	ApprovalDate	ReviewTime
1997	Anagrelide (Agrylin Capsules)	Roberts	1P	31JAN1996	14MAR1997	13.4
1997	Imiquod (Aldara Cream)	3M	1S	26JUL1996	27FEB1997	7.1
1997	Fomepizole (Antizol Injection)	Orphan Medical	1S	06DEC1996	04DEC1997	11.9
1997	Dolasetron mesylate (Anzemet)	HMR	1S	29SEP1995	11SEP1997	23.4
1997	Irbesartan (Avapro)	BMS	1S	26SEP1996	30SEP1997	12.1
1997	Cerivastatin sodium (Baycol)	Bayer	1S	26JUN1996	26JUN1997	12
1997	Fenoldopam mesylate (Corlopam)	Neurex	1S	25JUN1996	23SEP1997	15
1997	Bromfenac sodium capsules (Duract)	Wyeth-Ayerst	1S	30DEC1994	15JUL1997	30.5
1997	Emedastine difumarate (Emadine)	Alcon	1S	26MAR1996	29DEC1997	21.1
1997	Raloxifene HCl (Evista)	Eli Lilly	1P	09JUN1997	09DEC1997	6
1997	Toremifene citrate (Fareston)	Schering-Plough	1S	03JAN1995	29MAY1997	28.8
1997	Letrozole tablets (Femara)	Novartis	1S	25JUL1996	25JUL1997	12
1997	Tamsulosin HCl (Flomax)	Boehringer Ingel.	1S	15APR1996	15APR1997	12

This SAS table contains the raw data that we are using.

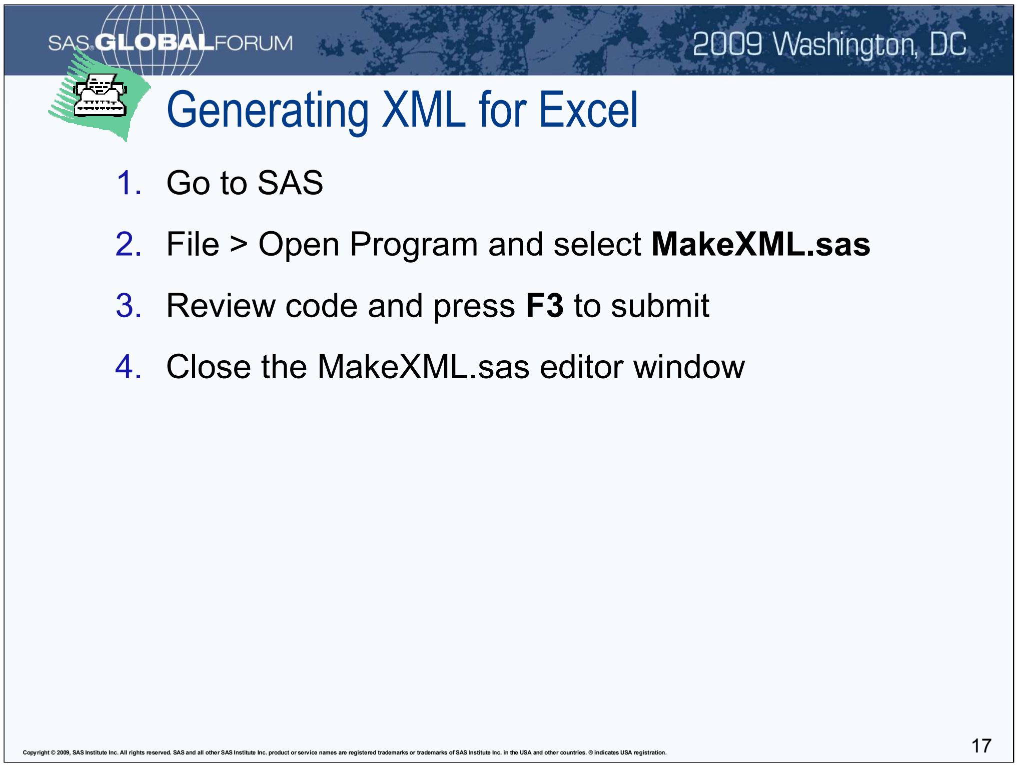
The column "Year" represents the year in which the drug was approved. One worksheet will be created for each unique value of this column.

The columns "Drug" and "Firm" are the drug name and the firm that owns the drug, respectively.

"Rating" is the FDA therapeutic classification, with "P" indicating that the drug offers a therapeutic advance, and "S" indicating that the drug is substantially equivalent to a previously-marketed drug. The "1" indicates that the drug is a new molecular entity (NME).

"SumbissionDate" and "ApprovalDate" are SAS date values representing when the drug was submitted and approved, respectively, and "ReviewTime" is the time, in months, between the submission and approval dates.

The SAS DATE9. format has been applied to the "SubmissionDate" and "ApprovalDate" columns.



Generating XML for Excel

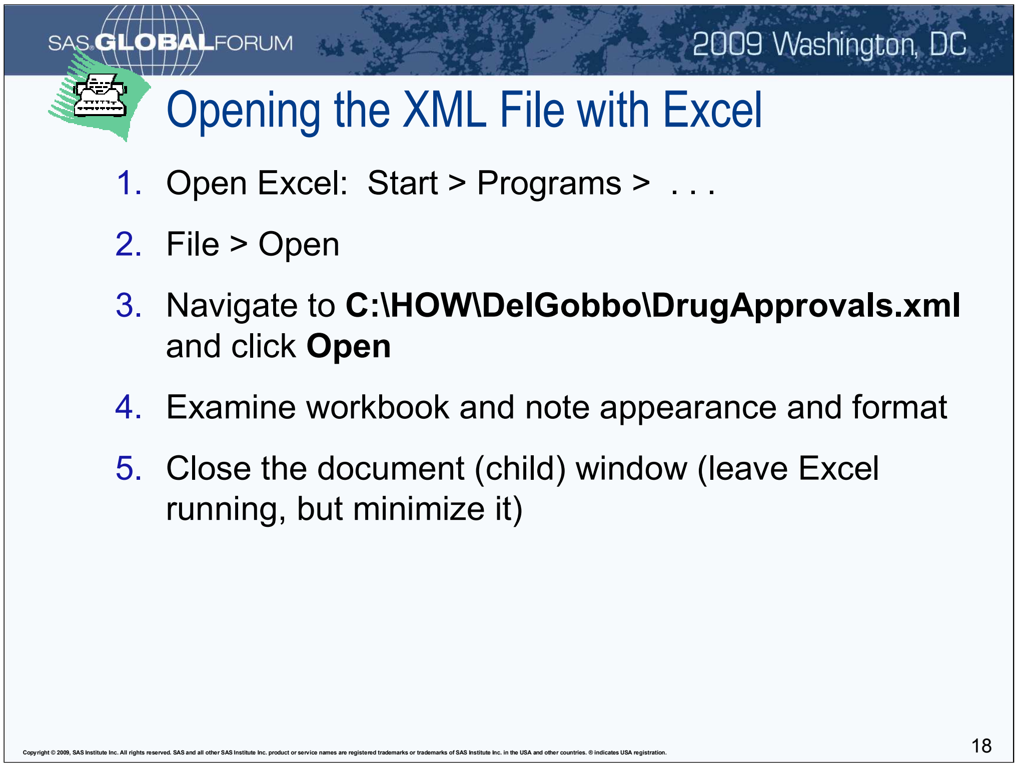
1. Go to SAS
2. File > Open Program and select **MakeXML.sas**
3. Review code and press **F3** to submit
4. Close the MakeXML.sas editor window

This is our first attempt to make an XML file that can be opened with Excel.

The ExcelXP tagset is used to generate XML output, and the XLsansPrinter style controls the appearance of the output.

The XML created by ODS is stored in a file named "DrugApprovals.xml", residing in the directory specified by the SAMPDIR macro variable.

By default, the ExcelXP tagset creates a new worksheet each time a SAS procedure creates new tabular output. And because BY statements create new tabular output for each BY group, and our data has BY groups for 5 years of data, our code outputs a total of 5 worksheets, one for each year.



Opening the XML File with Excel

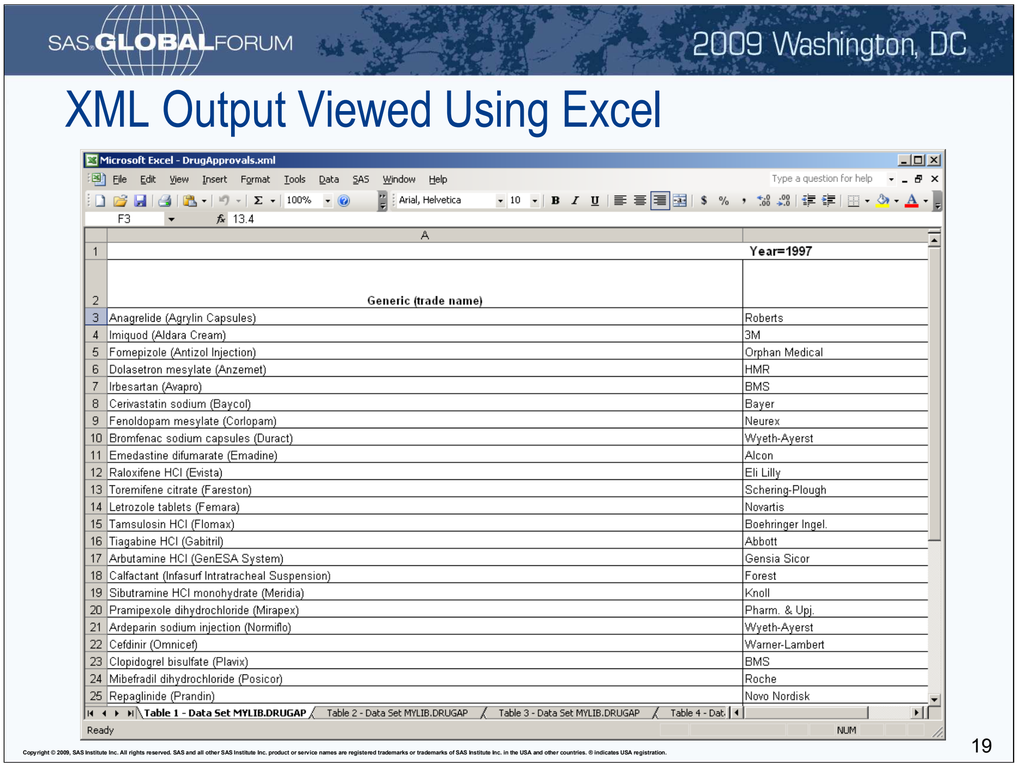
1. Open Excel: Start > Programs > . . .
2. File > Open
3. Navigate to **C:\HOW\DelGobbo\DrugApprovals.xml** and click **Open**
4. Examine workbook and note appearance and format
5. Close the document (child) window (leave Excel running, but minimize it)

The XML file created by ODS can now be opened with Microsoft Excel.

When Excel reads the XML file, it renders it as a multi-sheet workbook. Thus you can use all the editing and formatting features of Excel to modify the output and, if desired, save it as a native Excel workbook in binary format.

To save a copy of the file in Excel binary format using Excel 2002 or Excel 2003, select **File** ➨ **Save As** and then, from the **Save as type** drop-down list, select **Microsoft Excel Workbook (*.xls)**.

If you're using Excel 2007, click the Microsoft Office Button, and then select **Save As** ➨ **Excel Binary Workbook**.



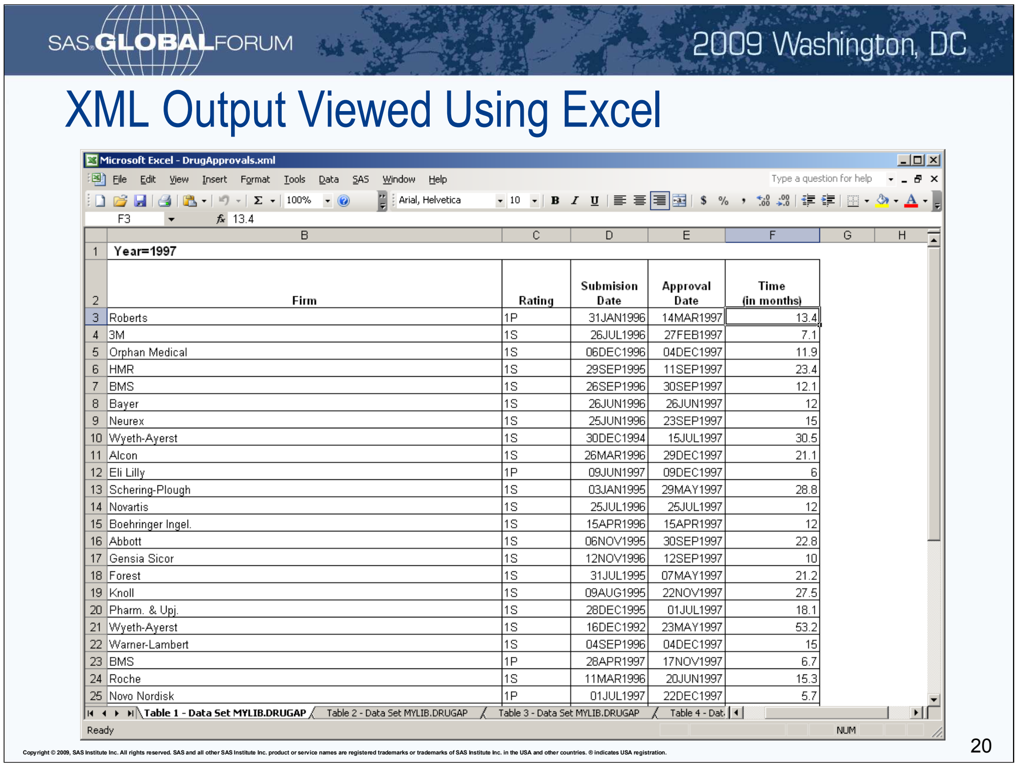
XML Output Viewed Using Excel

	A	
1		**Year=1997**
2	**Generic (trade name)**	
3	Anagrelide (Agrylin Capsules)	Roberts
4	Imiquod (Aldara Cream)	3M
5	Fomepizole (Antizol Injection)	Orphan Medical
6	Dolasetron mesylate (Anzemet)	HMR
7	Irbesartan (Avapro)	BMS
8	Cerivastatin sodium (Baycol)	Bayer
9	Fenoldopam mesylate (Corlopam)	Neurex
10	Bromfenac sodium capsules (Duract)	Wyeth-Ayerst
11	Emedastine difumarate (Emadine)	Alcon
12	Raloxifene HCl (Evista)	Eli Lilly
13	Toremifene citrate (Fareston)	Schering-Plough
14	Letrozole tablets (Femara)	Novartis
15	Tamsulosin HCl (Flomax)	Boehringer Ingel.
16	Tiagabine HCl (Gabitril)	Abbott
17	Arbutamine HCl (GenESA System)	Gensia Sicor
18	Calfactant (Infasurf Intratracheal Suspension)	Forest
19	Sibutramine HCl monohydrate (Meridia)	Knoll
20	Pramipexole dihydrochloride (Mirapex)	Pharm. & Upj.
21	Ardeparin sodium injection (Normiflo)	Wyeth-Ayerst
22	Cefdinir (Omnicef)	Warner-Lambert
23	Clopidogrel bisulfate (Plavix)	BMS
24	Mibefradil dihydrochloride (Posicor)	Roche
25	Repaglinide (Prandin)	Novo Nordisk

Table 1 - Data Set MYLIB.DRUGAP | Table 2 - Data Set MYLIB.DRUGAP | Table 3 - Data Set MYLIB.DRUGAP | Table 4 - Dat

Problems with the output:

1. The values in the "Rating" column should be centered, not left-justified.
2. The date values are displayed using a different format. If you double-click on the cell containing "31JAN1996" and press the ENTER key, Excel will change the date to "31-Jan-96".
3. Values in the "Time (in months)" column are not centered, are not all formatted with one decimal place, and are static values, instead of an Excel formula.
4. Standard BY group text (Year=1997) precedes the Excel tables, instead of custom titles.
5. Default worksheet names are used, instead of the value of the corresponding year.
6. Some of the columns are wider than they need to be, especially those containing the company and drug names.

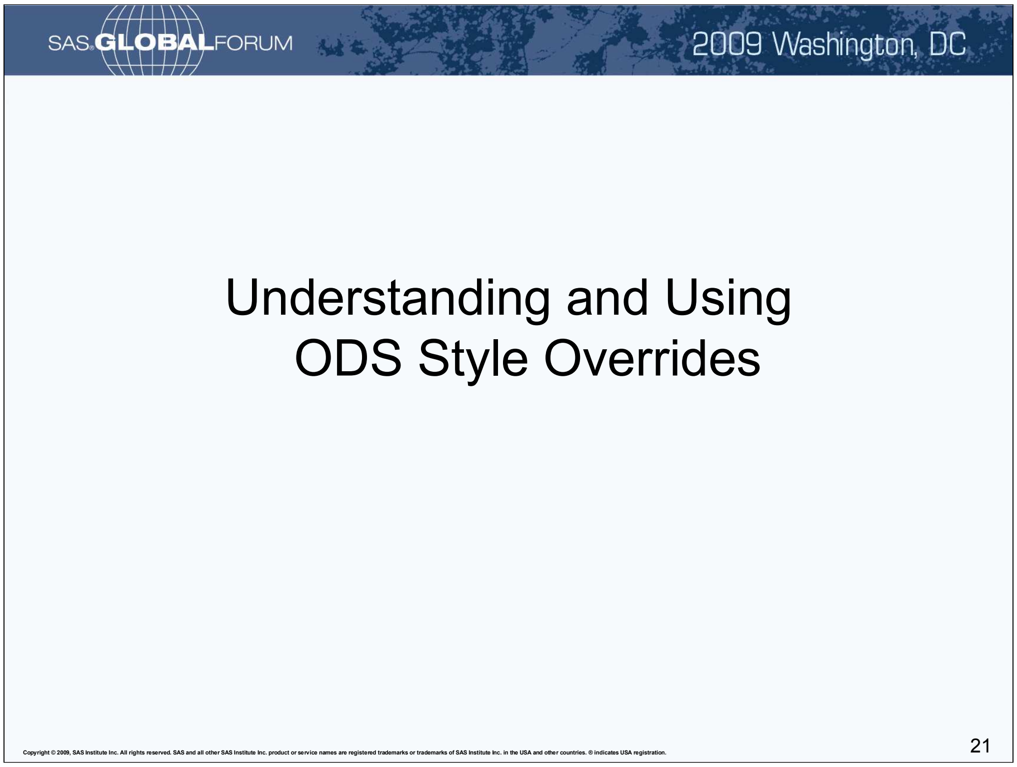

SAS GLOBALFORUM
2009 Washington, DC
Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Type a question for help
100%
Arial, Helvetica
10
F3
13.4
Table 1 - Data Set MYLIB.DRUGAP
Table 2 - Data Set MYLIB.DRUGAP
Table 3 - Data Set MYLIB.DRUGAP
Table 4 - Dat
Ready
NUM

XML Output Viewed Using Excel

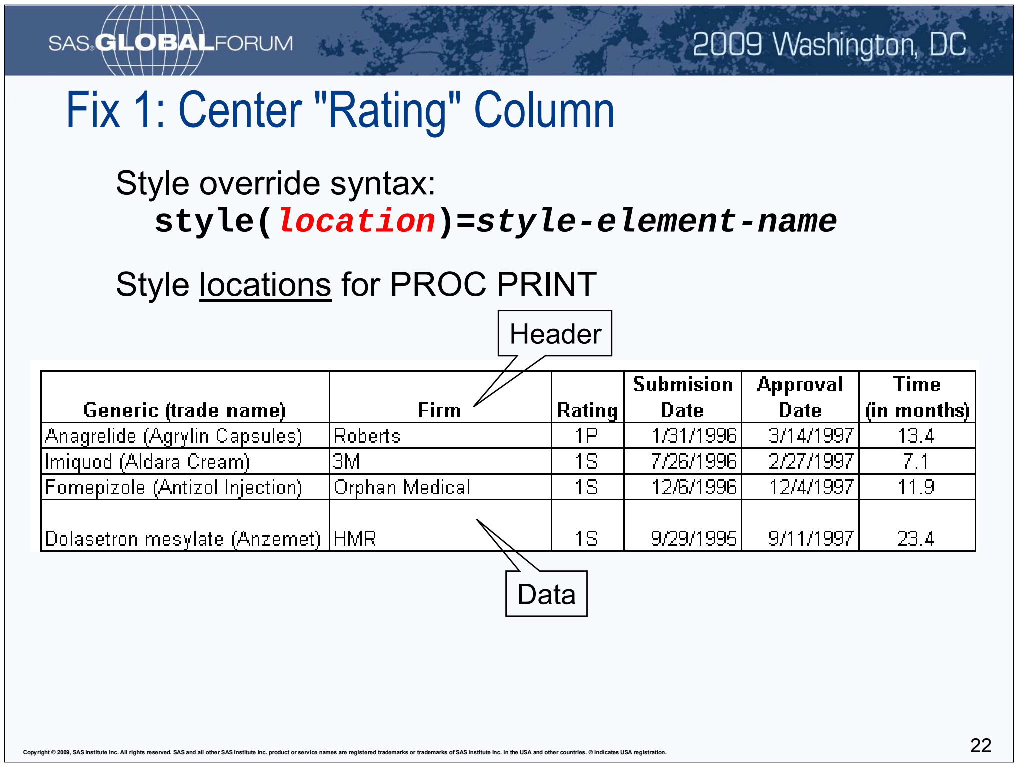
Year=1997

Firm	Rating	Submision Date	Approval Date	Time (in months)
Roberts	1P	31JAN1996	14MAR1997	13.4
3M	1S	26JUL1996	27FEB1997	7.1
Orphan Medical	1S	06DEC1996	04DEC1997	11.9
HMR	1S	29SEP1995	11SEP1997	23.4
BMS	1S	26SEP1996	30SEP1997	12.1
Bayer	1S	26JUN1996	26JUN1997	12
Neurex	1S	25JUN1996	23SEP1997	15
Wyeth-Ayerst	1S	30DEC1994	15JUL1997	30.5
Alcon	1S	26MAR1996	29DEC1997	21.1
Eli Lilly	1P	09JUN1997	09DEC1997	6
Schering-Plough	1S	03JAN1995	29MAY1997	28.8
Novartis	1S	25JUL1996	25JUL1997	12
Boehringer Ingel.	1S	15APR1996	15APR1997	12
Abbott	1S	06NOV1995	30SEP1997	22.8
Gensia Sicor	1S	12NOV1996	12SEP1997	10
Forest	1S	31JUL1995	07MAY1997	21.2
Knoll	1S	09AUG1995	22NOV1997	27.5
Pharm. & Upj.	1S	28DEC1995	01JUL1997	18.1
Wyeth-Ayerst	1S	16DEC1992	23MAY1997	53.2
Warner-Lambert	1S	04SEP1996	04DEC1997	15
BMS	1P	28APR1997	17NOV1997	6.7
Roche	1S	11MAR1996	20JUN1997	15.3
Novo Nordisk	1P	01JUL1997	22DEC1997	5.7

The remaining columns in the worksheet shown on the previous slide.



Understanding and Using ODS Style Overrides



Fix 1: Center "Rating" Column

Style override syntax:

`style(`*`location`*`)=`*`style-element-name`*

Style <u>locations</u> for PROC PRINT

Header

Generic (trade name)	Firm	Rating	Submision Date	Approval Date	Time (in months)
Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4

Data

The TEMPLATE procedure enables you to modify existing style elements and create new ones. Style overrides are used to apply these new style elements to specific parts of SAS output called *locations*.

Here you can see two locations that are pertinent to the PROC PRINT output: "Header" and "Data".

To change the appearance of the text in the column header for the "Firm" column, you would apply a style override to the "Header" location for this column:

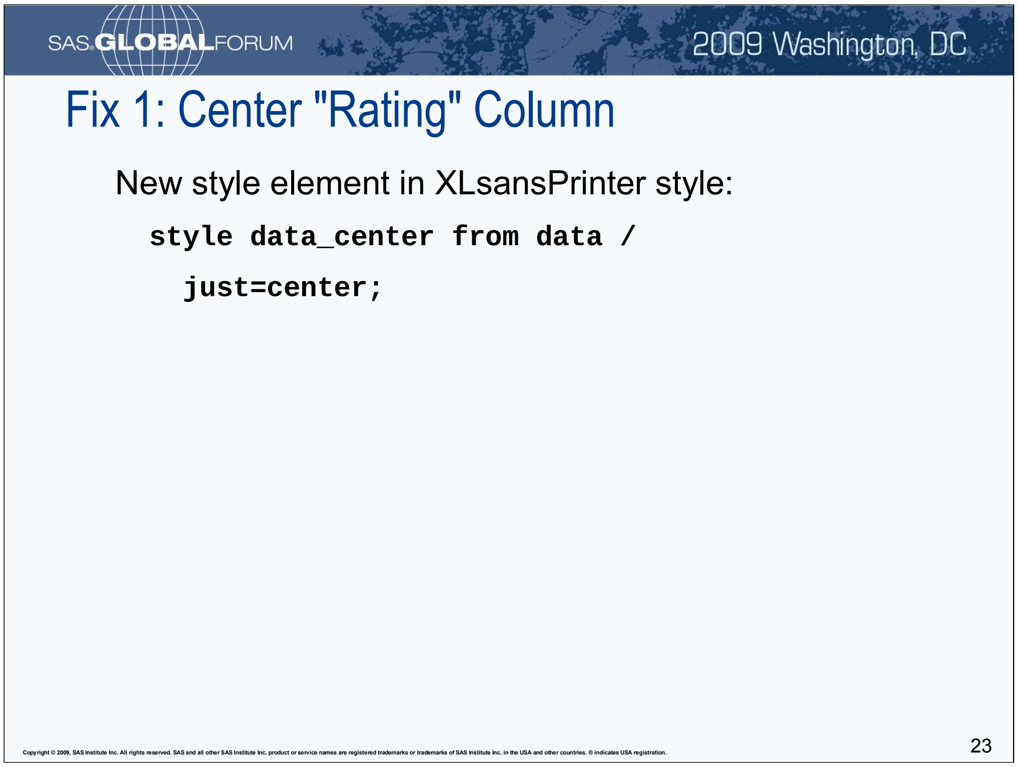
```
var Firm / style(Header)=style-element-name
```

Similarly, a style override applied to the "Data" location would affect the cells below the column headings:

```
var Firm / style(Data)=style-element-name
```

You can apply style overrides to more than one location. This code affects both the "Header" and "Data" locations:

```
var Firm / style(Header)=style-element-name
           style(Data)=style-element-name
```



SAS GLOBAL FORUM 2009 Washington, DC

Fix 1: Center "Rating" Column

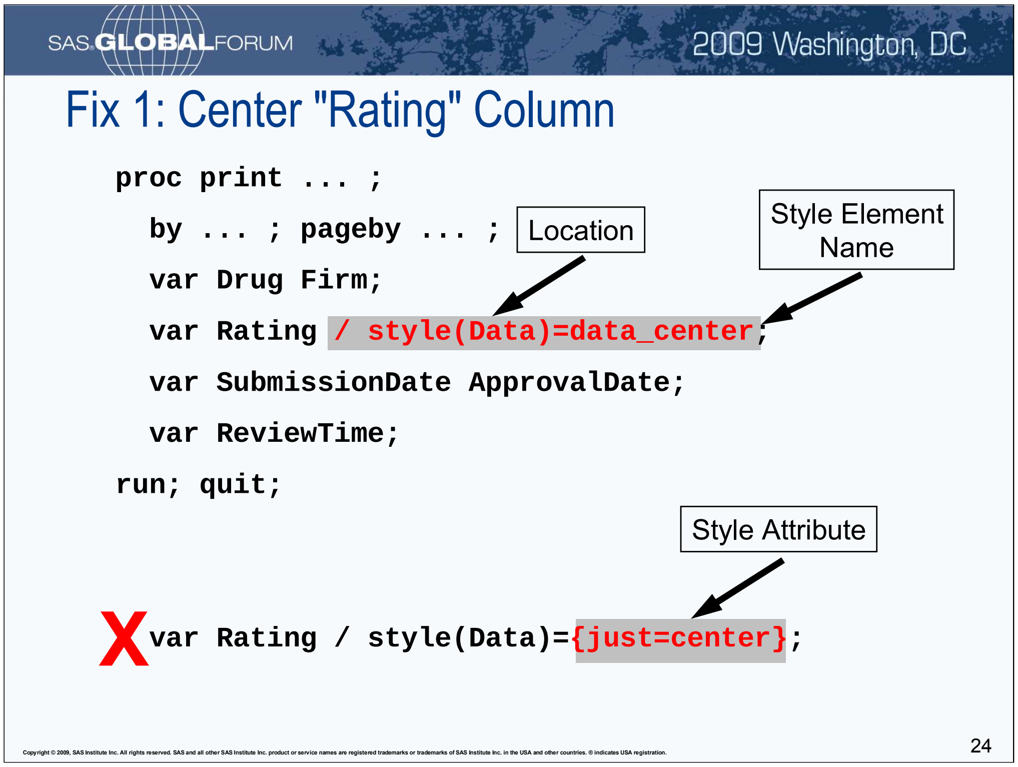
New style element in XLsansPrinter style:

```
style data_center from data /
  just=center;
```

As mentioned earlier, ODS styles control the appearance of your output. Here you see a subset of the code used to create the XLsansPrinter style. Refer to the "Setup.sas" file for a complete listing of the code.

The style element "data_center" is created based on the standard style element named "data". The "data_center" style element has all the same style attributes as its parent style element ("data"), except the justification attribute is set to "center".

We use this style element is used to control the appearance of the "Data" location of the "Rating" variable.



Fix 1: Center "Rating" Column

```
proc print ... ;
  by ... ; pageby ... ;
  var Drug Firm;
  var Rating / style(Data)=data_center;
  var SubmissionDate ApprovalDate;
  var ReviewTime;
run; quit;
```

Location

Style Element Name

Style Attribute

X
```
  var Rating / style(Data)={just=center};
```

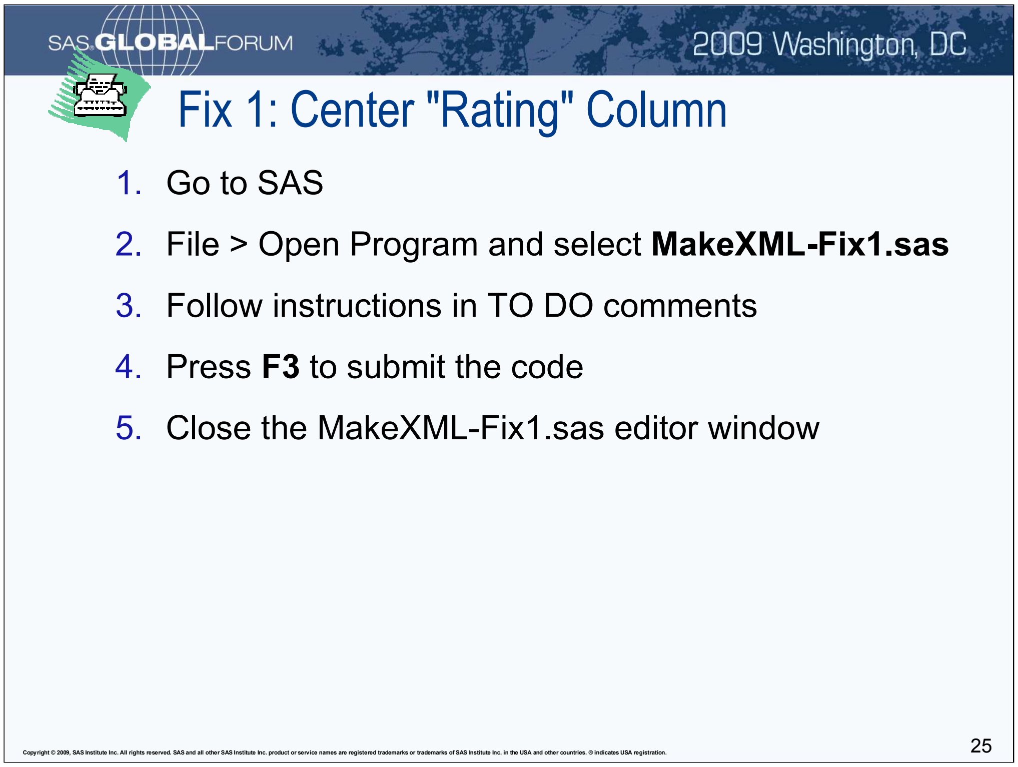
Adding this style override causes the text in the data cells for the "Rating" column to be centered.

The style element name, which contains the style attribute setting, is used

```
style(Data)=data_center
```

instead of the less efficient format, which uses the style attribute setting

```
style(Data)={just=center}
```


SAS GLOBAL FORUM
2009 Washington, DC

Fix 1: Center "Rating" Column

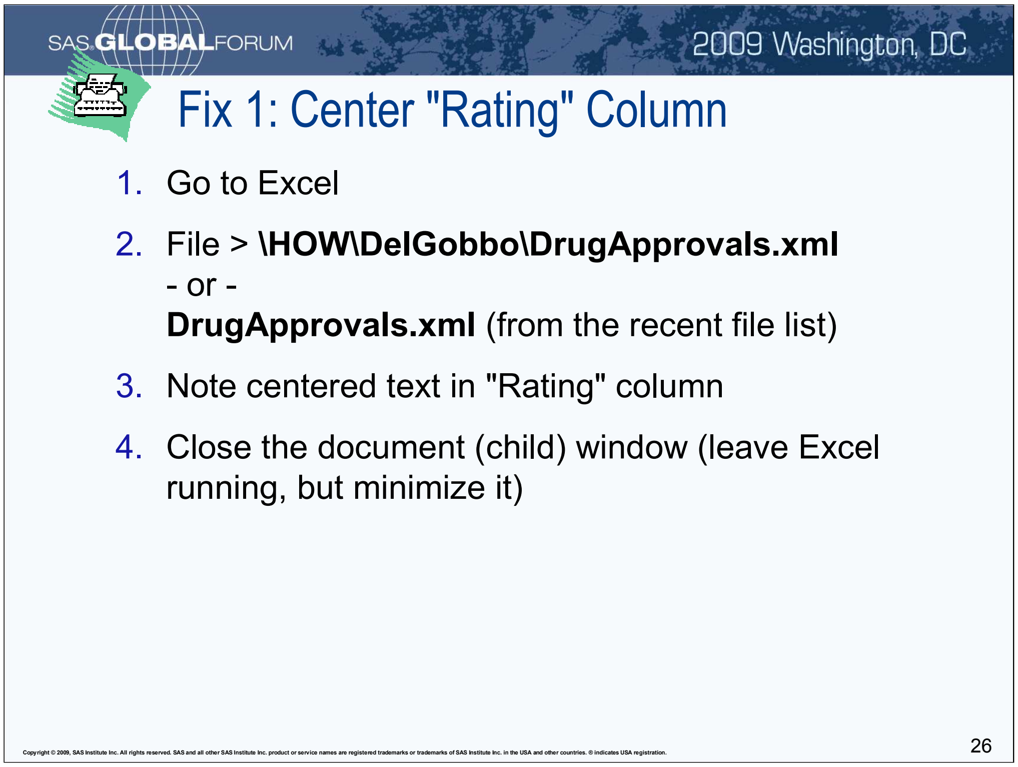
1. Go to SAS
2. File > Open Program and select **MakeXML-Fix1.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix1.sas editor window

TO DO:

Line 27: Add the style element named `data_center` to the style option.

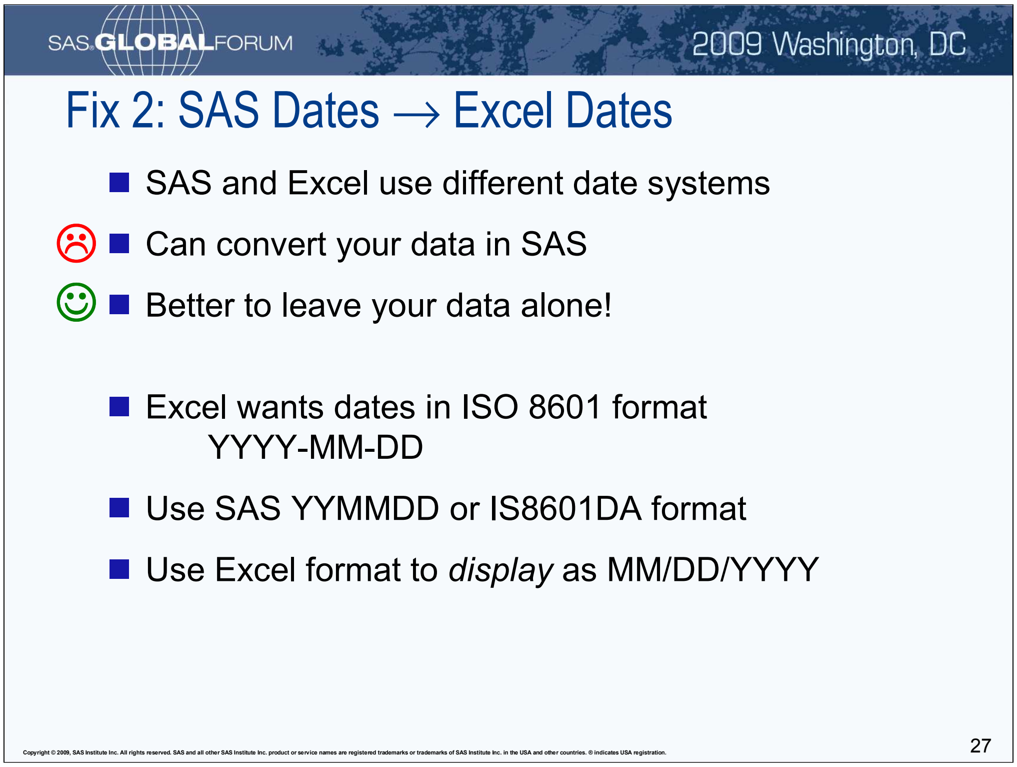
Solution:

```
var Rating / style(Data)=data_center;
```



Fix 1: Center "Rating" Column

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Note centered text in "Rating" column
4. Close the document (child) window (leave Excel running, but minimize it)



Fix 2: SAS Dates → Excel Dates

- SAS and Excel use different date systems
- ☹ Can convert your data in SAS
- ☺ Better to leave your data alone!

- Excel wants dates in ISO 8601 format
  YYYY-MM-DD
- Use SAS YYMMDD or IS8601DA format
- Use Excel format to *display* as MM/DD/YYYY

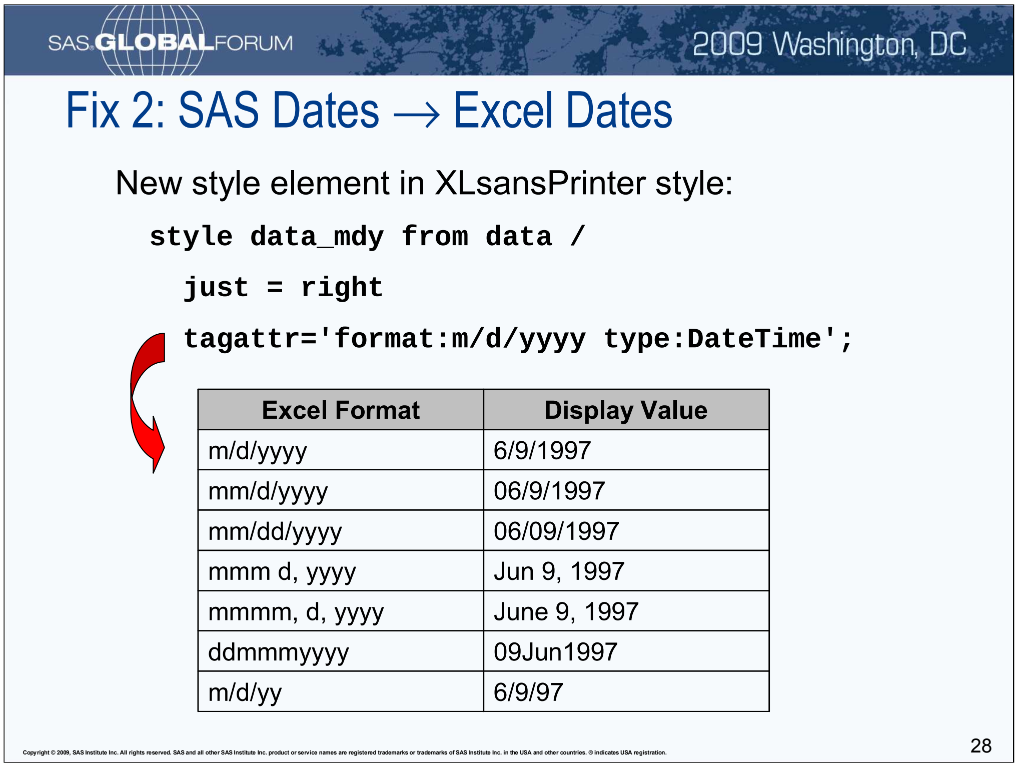
SAS and Microsoft Excel use different date systems. Consequently, you often encounter problems when Excel reads SAS output containing date values. For example, if you double-click on the cell containing "31JAN1996" and press the ENTER key, Excel will change the date to "31-Jan-96".

One way to correct this behavior is to write SAS code to convert SAS date values to Excel date values, but this approach is problematic because you must alter your original SAS data. While you can create a new view or SAS table that contains the new date values, this is a vector for errors and becomes inefficient as your data grows.

A better solution, one that does not require you to alter the underlying data, is to use a combination of SAS and Excel formats. First you specify a SAS format using a FORMAT statement, and then you specify an Excel format using a style override. The SAS format changes what is physically written into the XML file, and the Excel format changes the way the value is displayed.

Reference:

SAS Institute Inc. "Usage Note 11206: IS8601* FORMATS and INFORMATS for DATE, TIME, and DATETIME".
http://support.sas.com/kb/11/206.html



SAS GLOBAL FORUM 2009 Washington, DC

Fix 2: SAS Dates → Excel Dates

New style element in XLsansPrinter style:

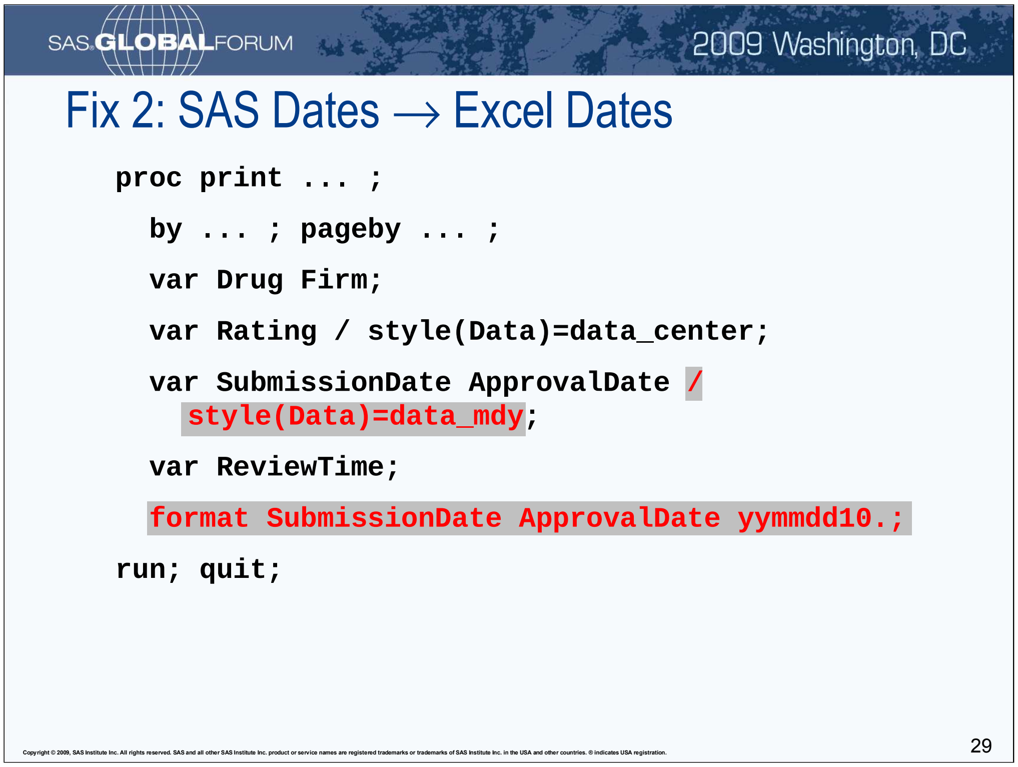
```
style data_mdy from data /
  just = right
  tagattr='format:m/d/yyyy type:DateTime';
```

Excel Format	Display Value
m/d/yyyy	6/9/1997
mm/d/yyyy	06/9/1997
mm/dd/yyyy	06/09/1997
mmm d, yyyy	Jun 9, 1997
mmmm, d, yyyy	June 9, 1997
ddmmmyyyy	09Jun1997
m/d/yy	6/9/97

The "XLsansPrinter" style contains the style element "data_mdy", which inherits all the attributes of the standard "data" style element, except that the text is right-justified. The "tagattr" attribute is used to apply an Excel format to the data, and also to instruct Excel to interpret the value as an *Excel* datetime value.

This table shows how the date June 9, 1997 is displayed in Excel using different formats, including the format used by the "data_mdy" style element ("m/d/yyyy").

To find out more about Excel date formats, type "display numbers as dates or times" into the Excel help system or refer to a book that covers this topic.



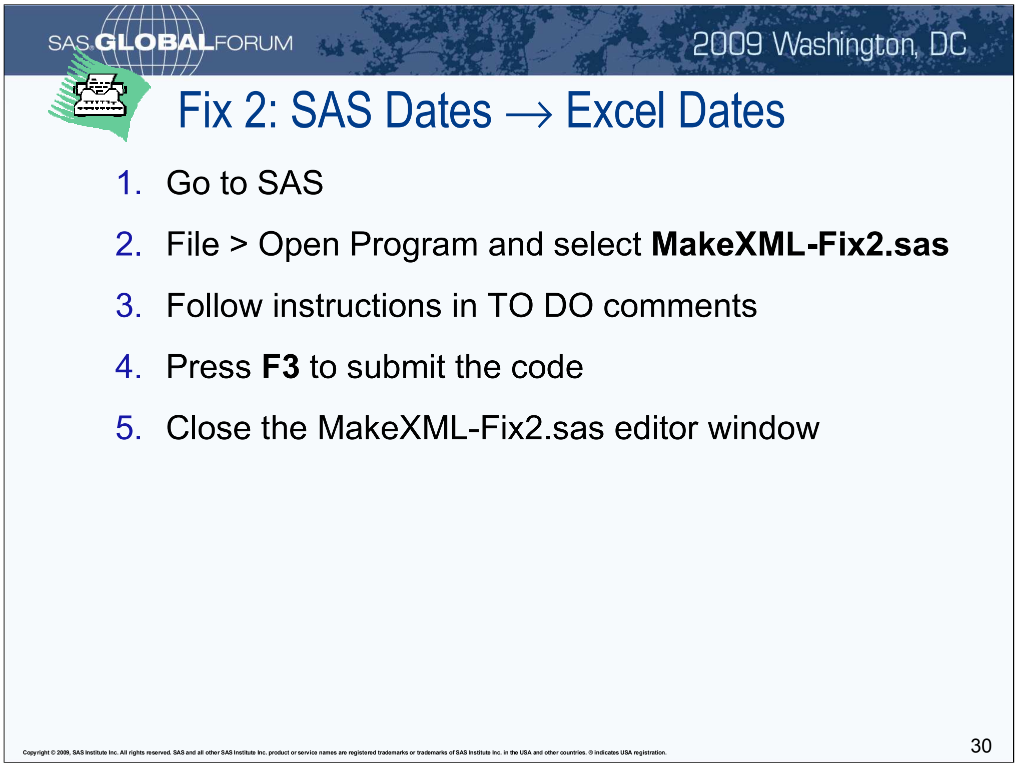
SAS GLOBAL FORUM 2009 Washington, DC

Fix 2: SAS Dates → Excel Dates

```
proc print ... ;
  by ... ; pageby ... ;
  var Drug Firm;
  var Rating / style(Data)=data_center;
  var SubmissionDate ApprovalDate /
    style(Data)=data_mdy;
  var ReviewTime;
  format SubmissionDate ApprovalDate yymmdd10.;
run; quit;
```

Because Excel expects dates to be represented using the ISO 8601 format (YYYY-MM-DD), SAS date values should be formatted using either the YYMMDD or IS8601DA format. We use the YYMMDD format for the "SubmissionDate" and "ApprovalDate" columns, which causes the values in the XML file to represented using the format YYYY-MM-DD.

The Excel format, applied as a result of using the style element "data_mdy", causes the date values to be *displayed* according to the format MM/DD/YYYY, using 2-digit values for the month and day only when necessary.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 2: SAS Dates → Excel Dates

1. Go to SAS
2. File > Open Program and select **MakeXML-Fix2.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix2.sas editor window

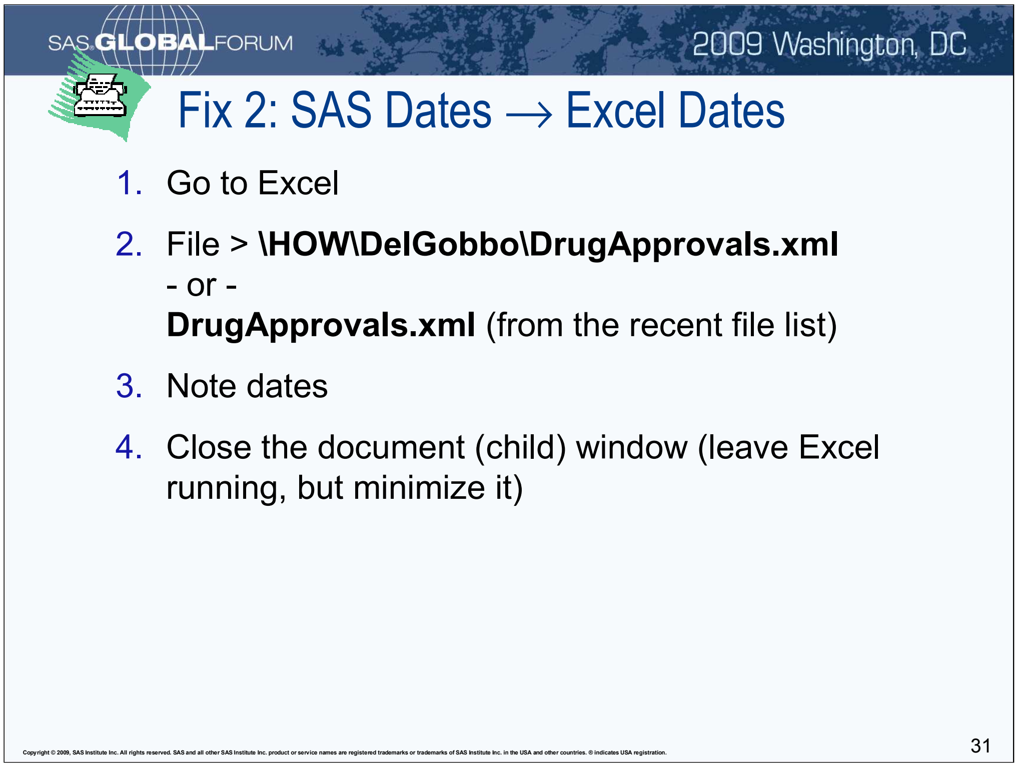
TO DO:

Line 32: Use the style element named `data_mdy` to control the appearance of the "Data" location of the SubmissionDate and ApprovalDate columns.

Line 34: Note the use of the SAS format for the SubmissionDate and ApprovalDate columns.

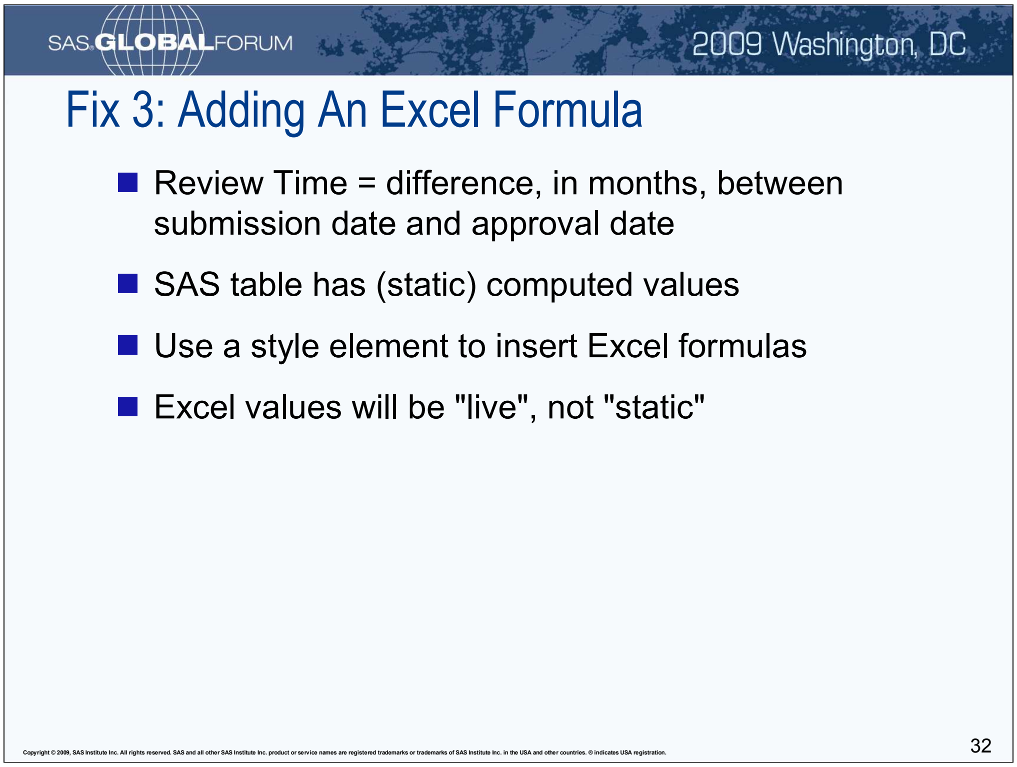
Solution:

```
var SubmissionDate ApprovalDate / style(Data)=data_mdy;
```



Fix 2: SAS Dates → Excel Dates

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Note dates
4. Close the document (child) window (leave Excel running, but minimize it)

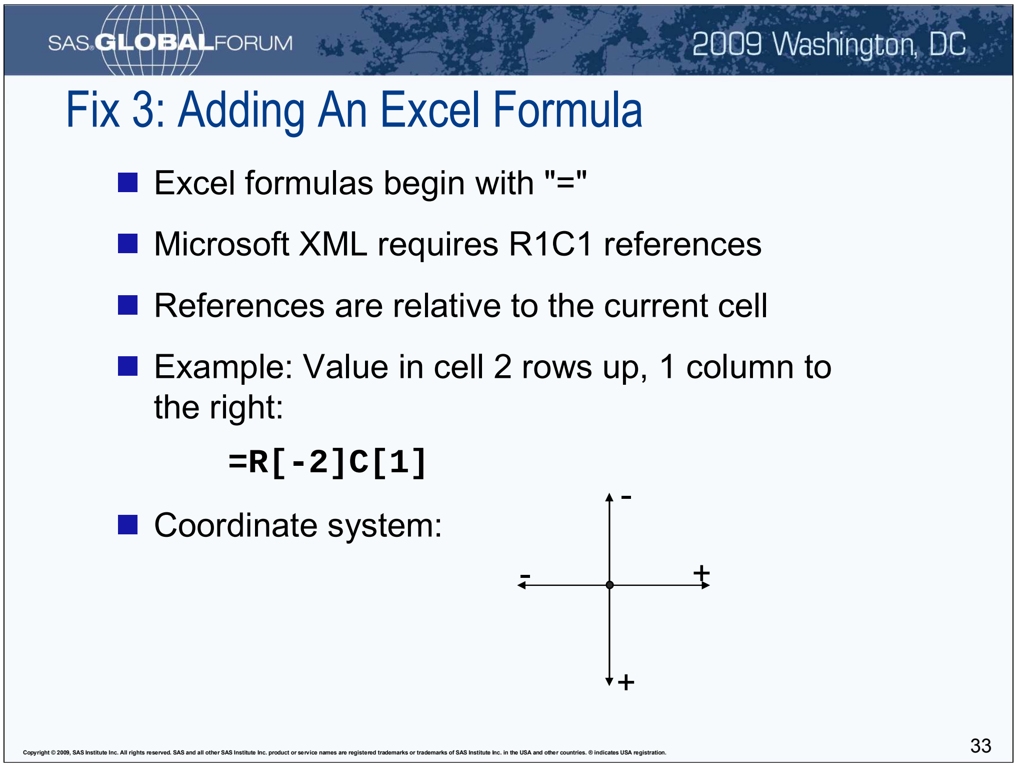


Fix 3: Adding An Excel Formula

- Review Time = difference, in months, between submission date and approval date
- SAS table has (static) computed values
- Use a style element to insert Excel formulas
- Excel values will be "live", not "static"

Calculating the review time is a simple matter of finding the difference between the date when the application was submitted, and the date when it was approved.

This value is already included in the SAS table, but the value results in "static" data in the Excel workbook. Instead, a style element is used to insert Excel the appropriate Excel formula into each cell, providing "live" data in the worksheets. That is, if the submission or approval date values are updated in a worksheet, the corresponding review time value is automatically recalculated by Excel.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 3: Adding An Excel Formula

- Excel formulas begin with "="
- Microsoft XML requires R1C1 references
- References are relative to the current cell
- Example: Value in cell 2 rows up, 1 column to the right:

  `=R[-2]C[1]`
- Coordinate system:

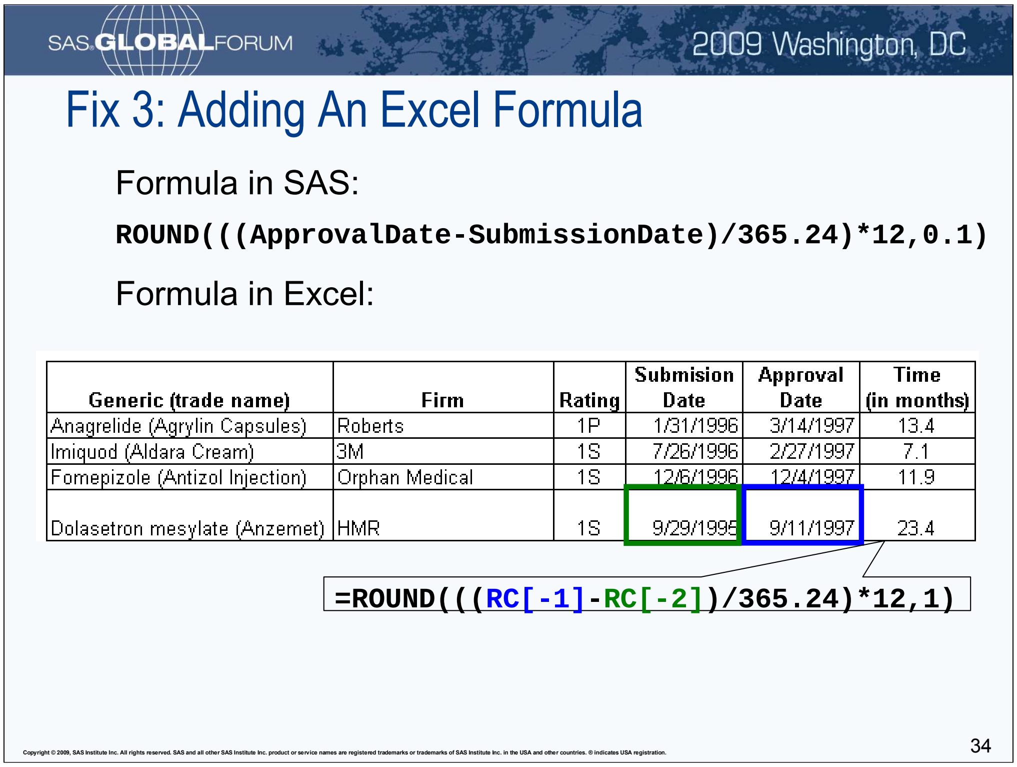
Excel formulas begin with an equal sign ("="), and usually contain references to other cells in a worksheet. There are two ways to reference cells: the A1 style and the R1C1 style. Although the A1 style is more widely used, the Microsoft XML Spreadsheet Specification requires that R1C1-style references be used when building worksheets with XML because "they are significantly easier to parse and generate than A1-style formulas".

R1C1-style references are relative to the current cell. Thus, the formula `=R[-2]C[1]` refers to the value in the cell 2 rows up and 1 column to the right of the cell containing the formula.

For more detailed information about the Excel R1C1 reference style, type "about cell and range references" into the Excel help system, or refer to a book that covers this topic.

Reference:

Microsoft Corporation. "XML Spreadsheet Reference". http://msdn2.microsoft.com/en-us/library/aa140066(office.10).aspx.



Fix 3: Adding An Excel Formula

Formula in SAS:

```
ROUND(((ApprovalDate-SubmissionDate)/365.24)*12,0.1)
```

Formula in Excel:

Generic (trade name)	Firm	Rating	Submision Date	Approval Date	Time (in months)
Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4

```
=ROUND(((RC[-1]-RC[-2])/365.24)*12,1)
```

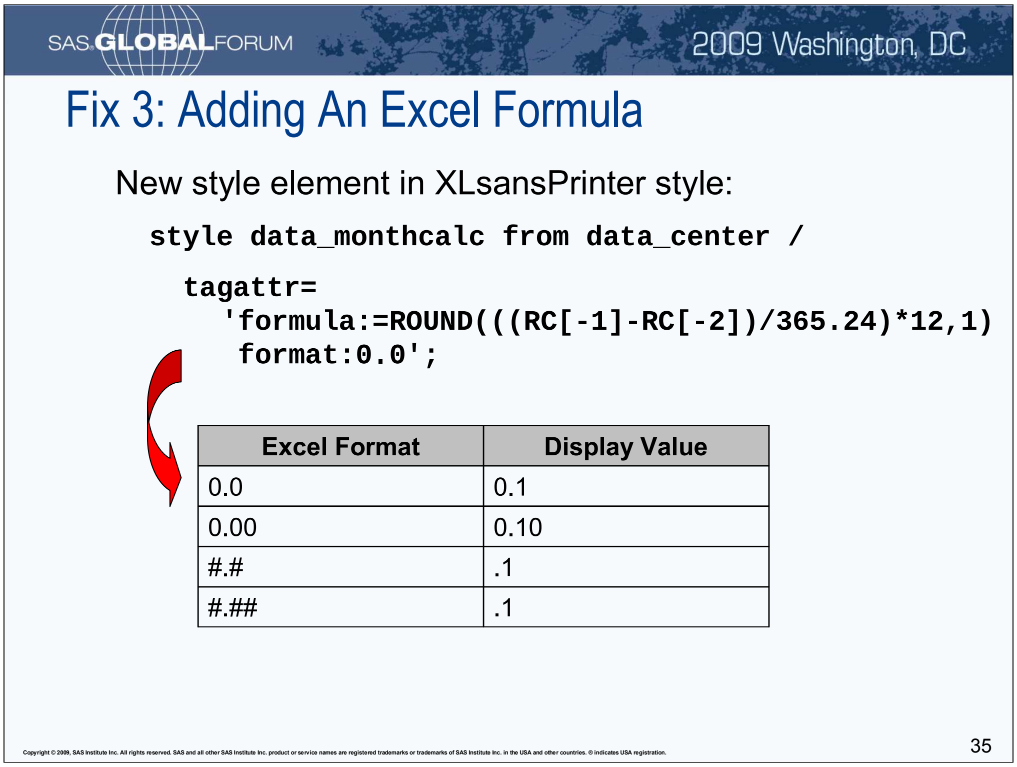
The difference between the approval date and the review date is represented in days. We divide the difference by 365.24 to come up with the number of years, then multiply by 12 to get the final value, specified in months. The final value is rounded to 1 decimal place. Using SAS, the formula would be:

```
ROUND(((ApprovalDate-SubmissionDate)/365.24)*12,0.1)
```

In Excel, the formula is very similar, except for the R1C1 nomenclature. Because the "ApprovalDate" column is one column to the left of the column containing the formula, it is represented as "RC[-1]". Similarly, the "SubmissionDate" column is represented as "RC[-2]". We do not need to indicate an index for the row for the Excel formula because all calculations are carried out within the row containing the formula.

While the R1C1 style format is used to store formulas in the XML file, the A1 style is displayed when viewing the file with Excel. That is, the R1C1 style formula shown above might be displayed in the Excel formula bar as:

```
=ROUND(((E5-D5)/365.24)*12,1)
```



SAS GLOBAL FORUM 2009 Washington, DC

Fix 3: Adding An Excel Formula

New style element in XLsansPrinter style:

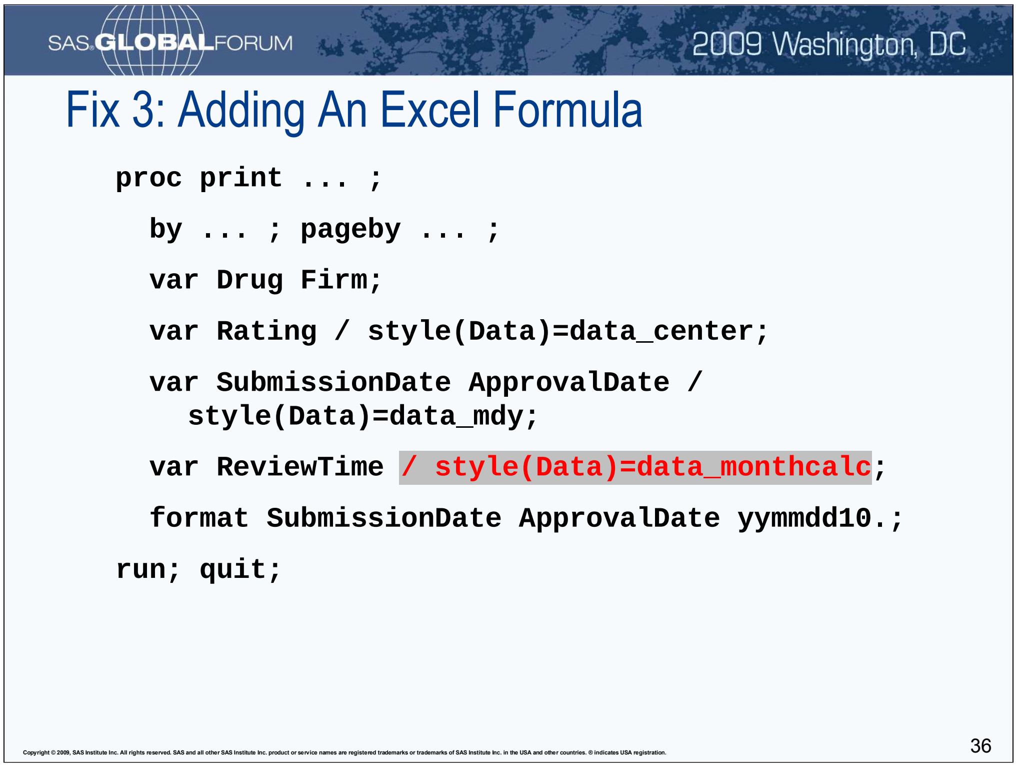
```
style data_monthcalc from data_center /

  tagattr=
    'formula:=ROUND(((RC[-1]-RC[-2])/365.24)*12,1)
     format:0.0';
```

Excel Format	Display Value
0.0	0.1
0.00	0.10
#.#	.1
#.##	.1

Up to this point we have used the "tagattr" style attribute to apply an Excel format and to specify the Excel data type. This attribute can also be used to assign an Excel formula to all the cells in a column. Here, the "tagattr" style attribute is used to apply both an Excel formula and an Excel format.

The table shows how the value 1/10 would be displayed in Excel using different formats, including the format used by the "data_monthcalc" style element ("0.0"). The "#" in an Excel format displays any numeric digit or digits, excluding insignificant zeroes, and the "0" instructs Excel to display a numeric digit or digits, including insignificant zeroes.

For more information about Excel formats, type "create a custom number format" into the Excel help system or refer to a book that covers this topic. Similarly, for more information about Excel formulas, type "create a formula" into the Excel help system.

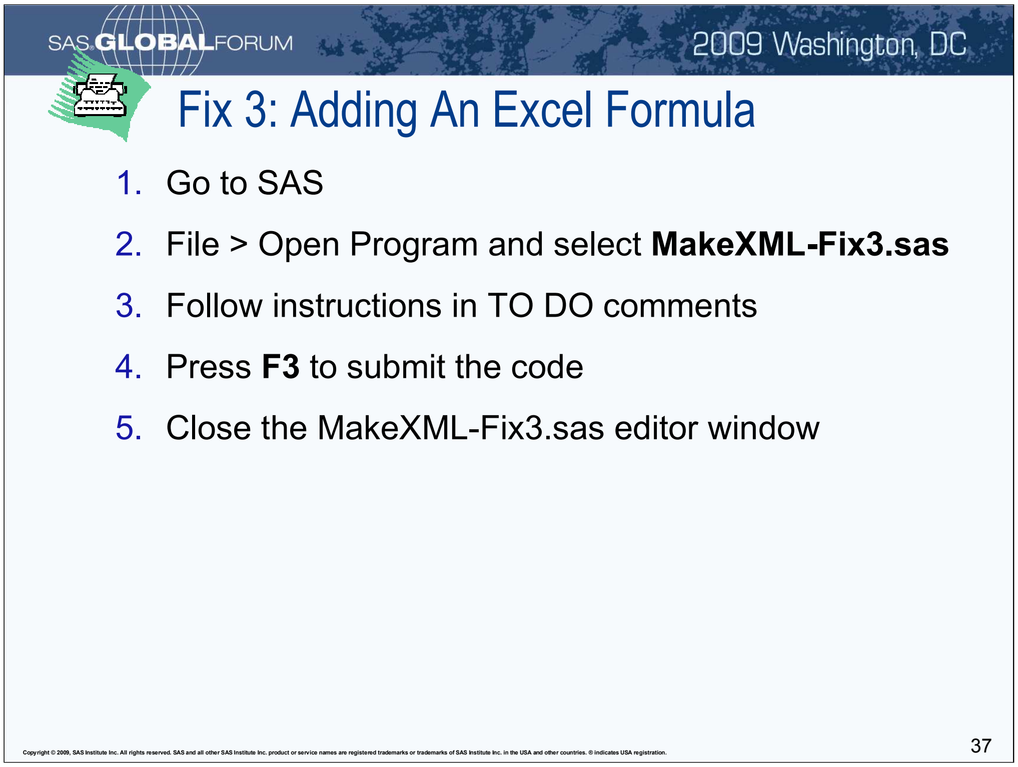


SAS GLOBAL FORUM 2009 Washington, DC

Fix 3: Adding An Excel Formula

```
proc print ... ;
  by ... ; pageby ... ;
  var Drug Firm;
  var Rating / style(Data)=data_center;
  var SubmissionDate ApprovalDate /
    style(Data)=data_mdy;
  var ReviewTime / style(Data)=data_monthcalc;
  format SubmissionDate ApprovalDate yymmdd10.;
run; quit;
```

Using the "data_monthcalc" style element for the "Data" location of the "ReviewTime" column causes the Excel formula and format to be applied.



Fix 3: Adding An Excel Formula

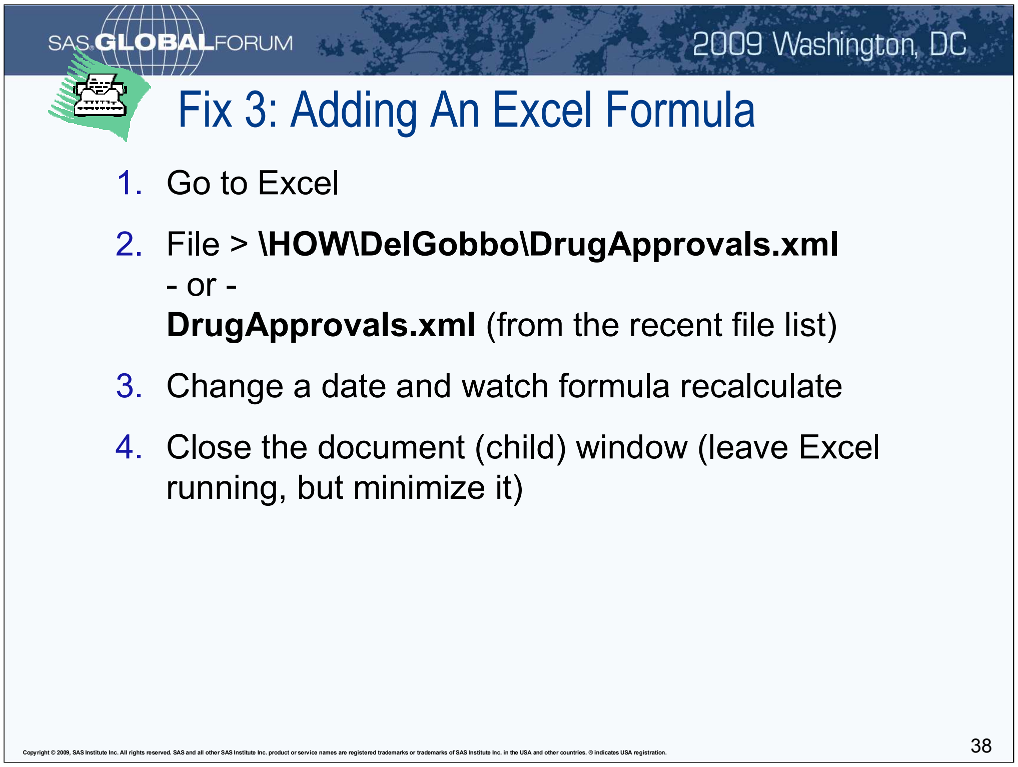
1. Go to SAS
2. File > Open Program and select **MakeXML-Fix3.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix3.sas editor window

TO DO:

Line 30: Use the style element named `data_monthcalc` to control the appearance of the "Data" location of the ReviewTime column.

Solution:

```
var ReviewTime / style(Data)=data_monthcalc;
```



SAS GLOBAL FORUM
2009 Washington, DC

Fix 3: Adding An Excel Formula

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Change a date and watch formula recalculate
4. Close the document (child) window (leave Excel running, but minimize it)

Exercise: Working with Excel Formats in "Setup.sas"

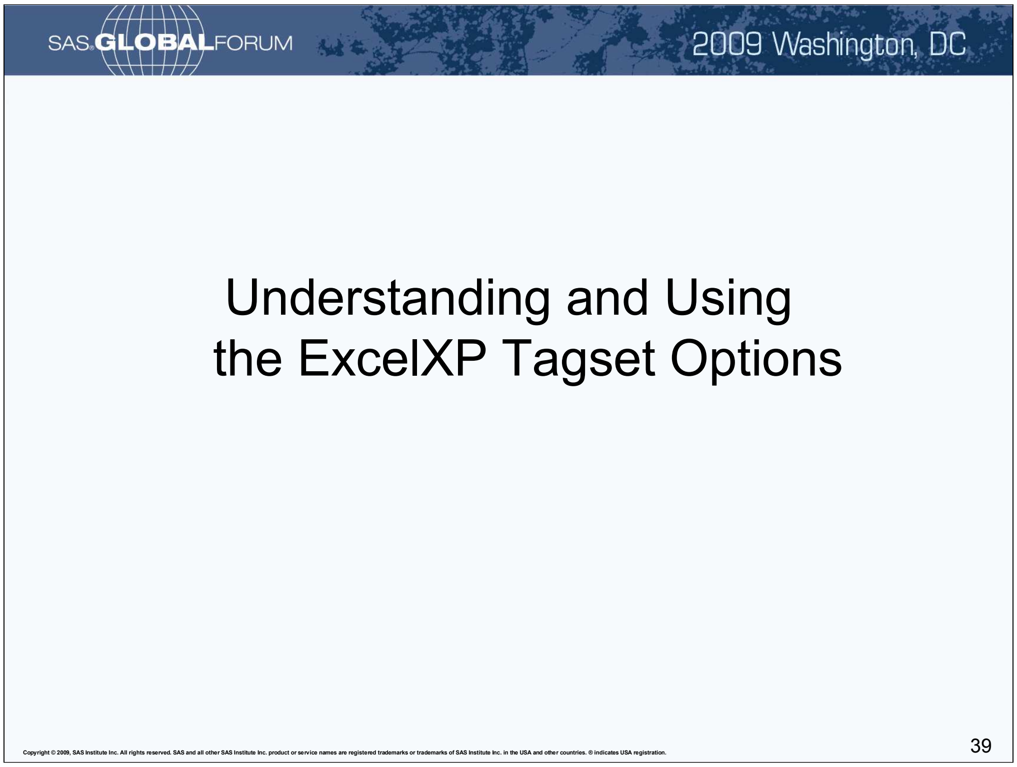
1. Change the Excel format in "Setup.sas", and rerun the code

```
style data_monthcalc from data_center / tagattr='formula: ...
 format:0.0';
```

2. Rerun "MakeXML-Fix3.sas"
3. View new output using Excel

Sample Excel formats to try, and their effect on the raw value 13.4:

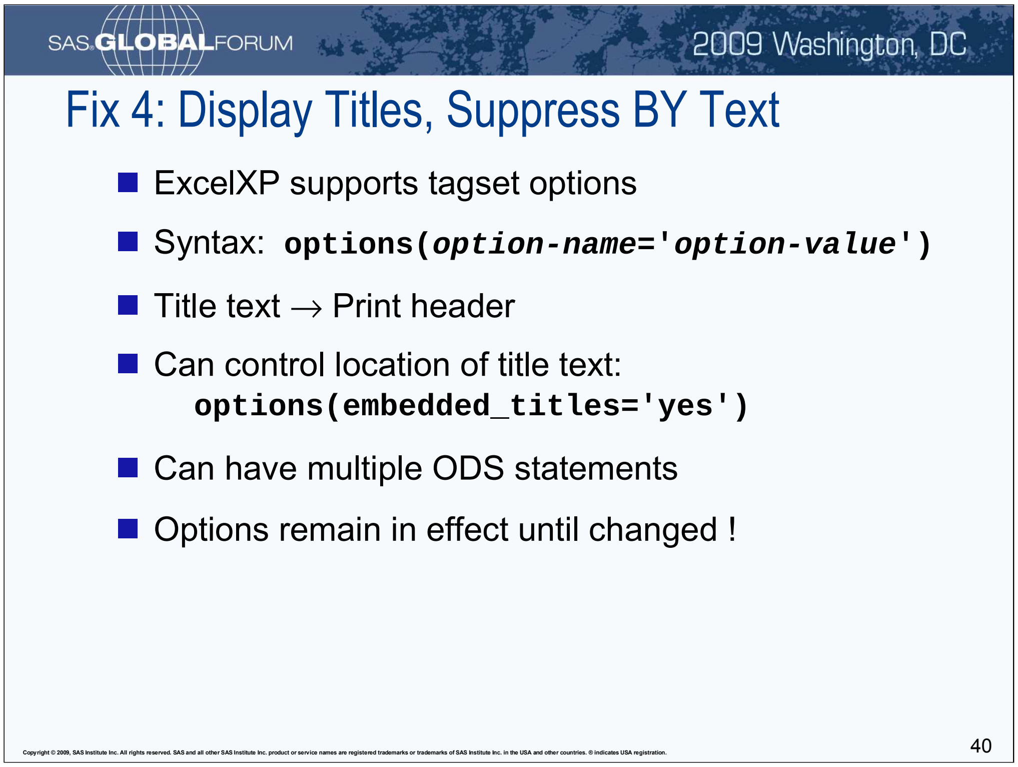
Excel Format	Display Value
Currency	$13.40
Percent	1340.00%
Scientific	1.34E+01
#	13
00000.00	00013.40
00000	00013



SAS GLOBAL FORUM

2009 Washington, DC

Understanding and Using the ExcelXP Tagset Options



SAS GLOBAL FORUM 2009 Washington, DC

Fix 4: Display Titles, Suppress BY Text

- ExcelXP supports tagset options
- Syntax: `options(option-name='option-value')`
- Title text → Print header
- Can control location of title text:
  `options(embedded_titles='yes')`
- Can have multiple ODS statements
- Options remain in effect until changed !

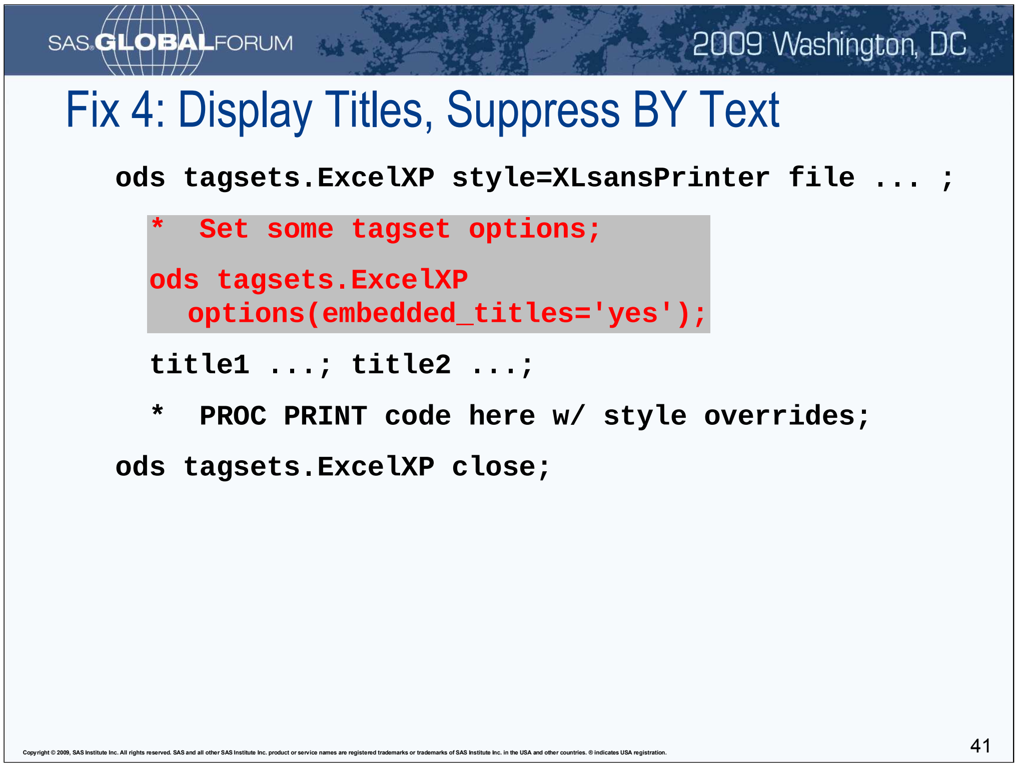
The ExcelXP tagset supports several options that control both the appearance and functionality of the Excel workbook. Many of these tagset options are simply tied directly to existing Excel options or features. Tagset options are specified on the ODS statement using the OPTIONS keyword, as shown above.

BY line text such as "Year=1997" precedes the tables in our worksheets. Instead, we want the text of the two TITLE statements to precede the tables.

By default, SAS titles and footnotes appear as Excel print headers and print footers, which are displayed only when the Excel document is printed. You can confirm this by viewing the Excel "Page Setup" dialog.

To include the titles in the worksheets, use the EMBEDDED_TITLES tagset option.

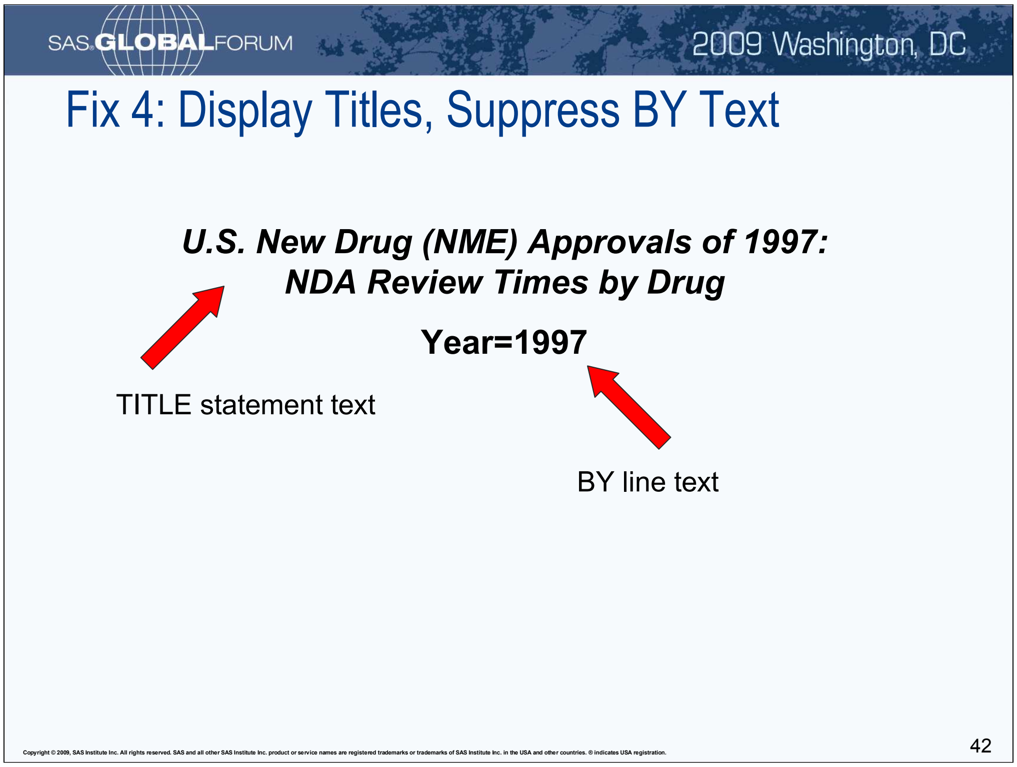
IMPORTANT NOTE: Tagset options remain in effect until they are changed !



Fix 4: Display Titles, Suppress BY Text

```
ods tagsets.ExcelXP style=XLsansPrinter file ... ;
  *  Set some tagset options;
  ods tagsets.ExcelXP
    options(embedded_titles='yes');
  title1 ...; title2 ...;
  *  PROC PRINT code here w/ style overrides;
ods tagsets.ExcelXP close;
```

The tagset options could have been added to the original ODS statement. Instead, a new statement was issued in order to increase readability of the slide and the SAS code.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 4: Display Titles, Suppress BY Text

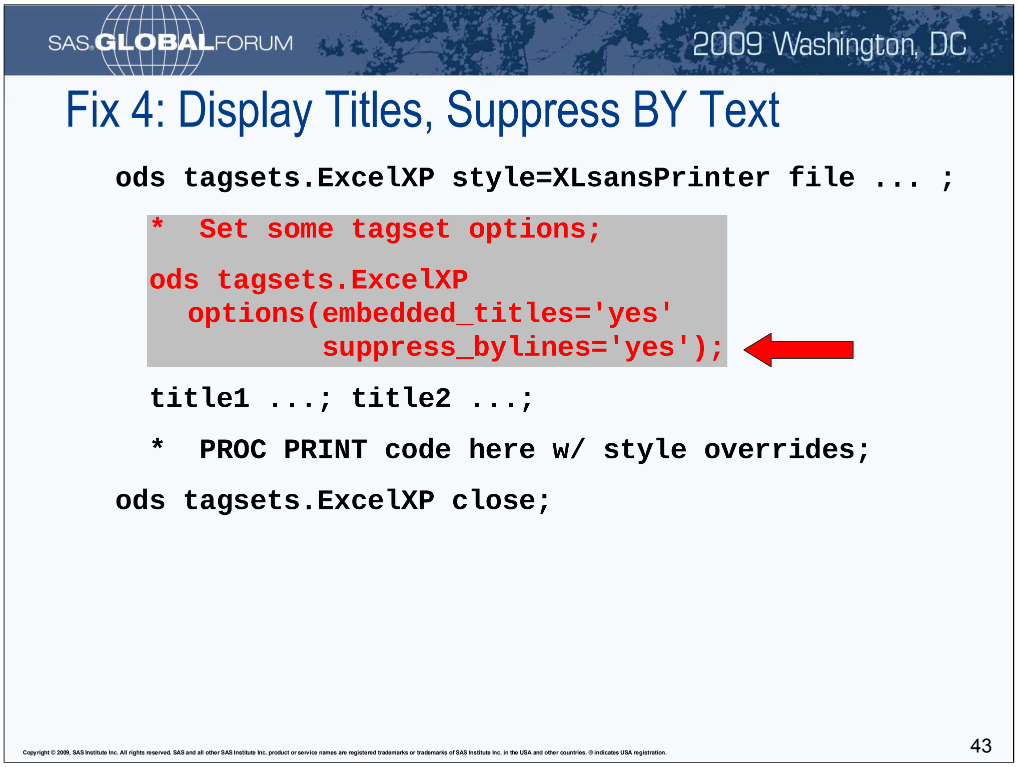
U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug

Year=1997

TITLE statement text

BY line text

After submitting the code on the previous slide and viewing the output using Excel, you will see that the TITLE statement text appears in the body of the worksheets. However, the BY line text is still displayed, too.


SAS GLOBAL FORUM
2009 Washington, DC

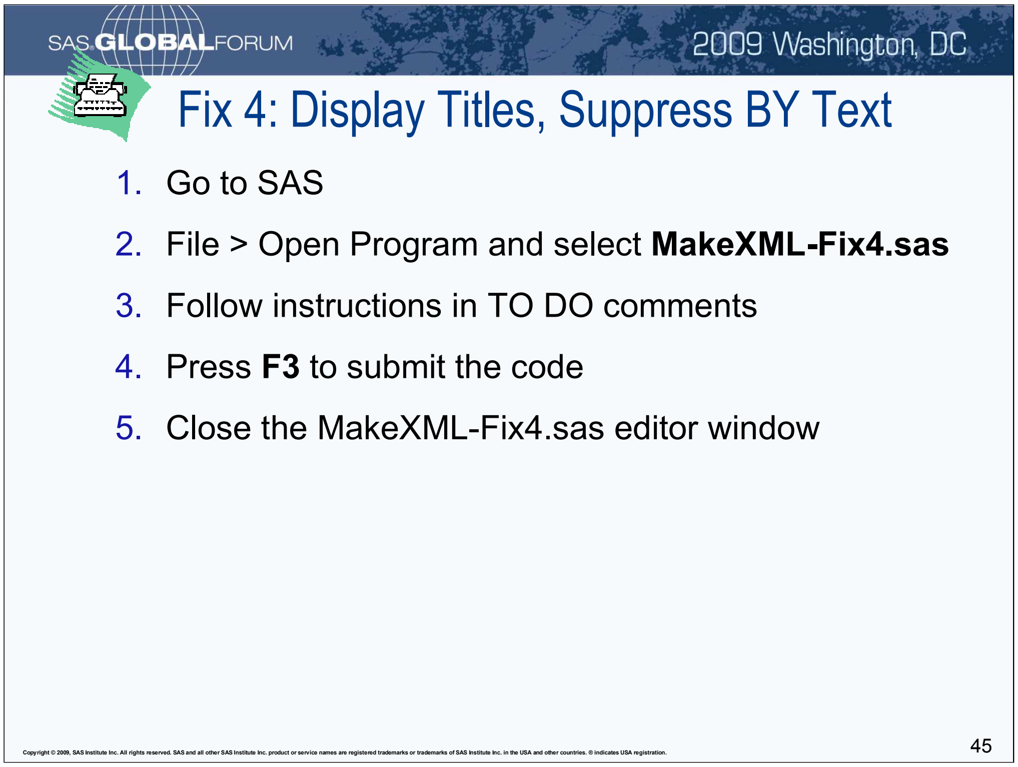
Fix 4: Display Titles, Suppress BY Text

```
ods tagsets.ExcelXP style=XLsansPrinter file ... ;

  *  Set some tagset options;

  ods tagsets.ExcelXP
    options(embedded_titles='yes'
            suppress_bylines='yes');

  title1 ...; title2 ...;

  *  PROC PRINT code here w/ style overrides;

ods tagsets.ExcelXP close;
```

The SUPPRESS_BYLINES option is used to prevent the BY line text from being included in the worksheets.

Fix 4: Display Titles, Suppress BY Text

U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug



SAS GLOBAL FORUM
2009 Washington, DC

Fix 4: Display Titles, Suppress BY Text

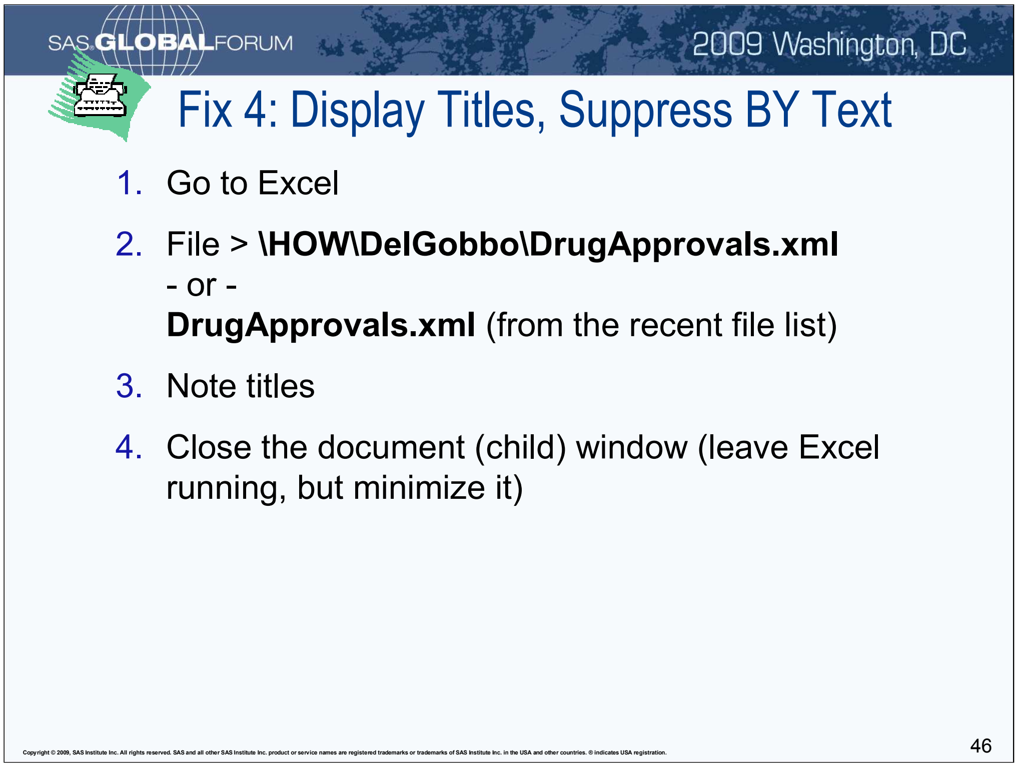
1. Go to SAS
2. File > Open Program and select **MakeXML-Fix4.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix4.sas editor window

TO DO:

Line 19: Supply the tagset options `embedded_titles='yes'` and `suppress_bylines='yes'`

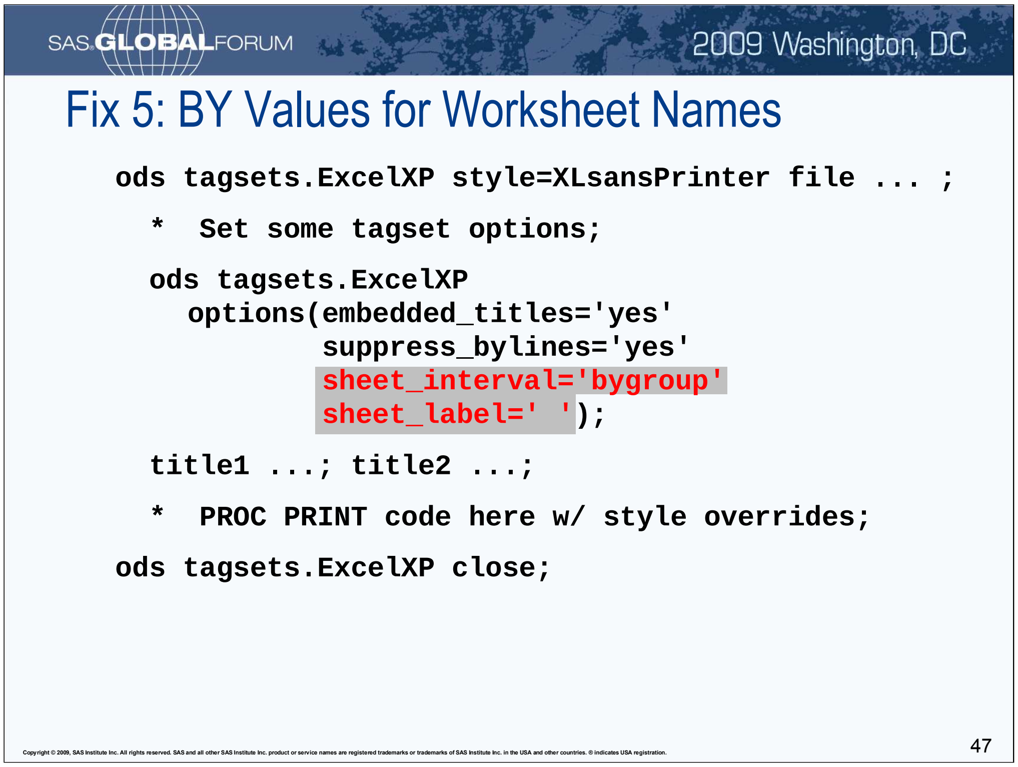
Solution:

```
ods tagsets.ExcelXP options(embedded_titles='yes',
                            suppress_bylines='yes');
```



Fix 4: Display Titles, Suppress BY Text

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Note titles
4. Close the document (child) window (leave Excel running, but minimize it)

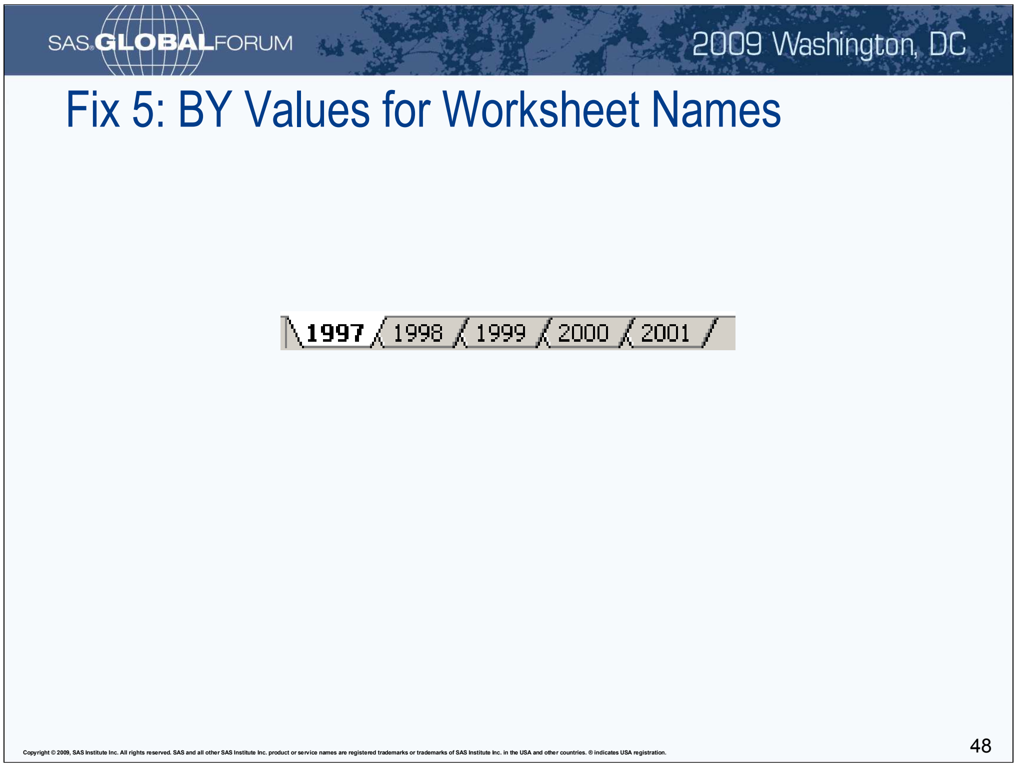


Fix 5: BY Values for Worksheet Names

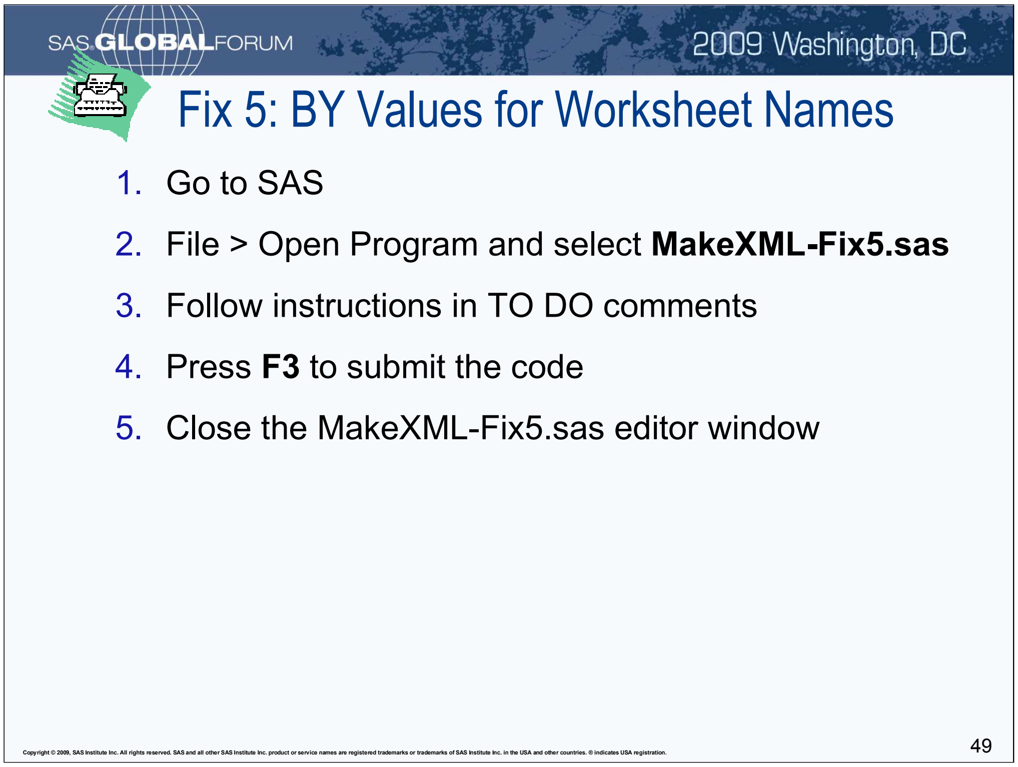
```
ods tagsets.ExcelXP style=XLsansPrinter file ... ;
  *  Set some tagset options;
  ods tagsets.ExcelXP
    options(embedded_titles='yes'
           suppress_bylines='yes'
           sheet_interval='bygroup'
           sheet_label=' ');
  title1 ...; title2 ...;
  *  PROC PRINT code here w/ style overrides;
ods tagsets.ExcelXP close;
```

ODS generates a unique name for each worksheet, as required by Microsoft Excel. These names are generally unappealing. There are, however, several tagset options that you can use to alter the names of the worksheets.

The SHEET_INTERVAL option controls the interval at which SAS output is placed into worksheets, and the SHEET_LABEL option is used to specify the prefix to use for the worksheet names. When used together, as shown here, the current value of the first BY group variable will be used as the worksheet name.



Fix 5: BY Values for Worksheet Names



SAS GLOBAL FORUM 2009 Washington, DC

Fix 5: BY Values for Worksheet Names

1. Go to SAS
2. File > Open Program and select **MakeXML-Fix5.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix5.sas editor window

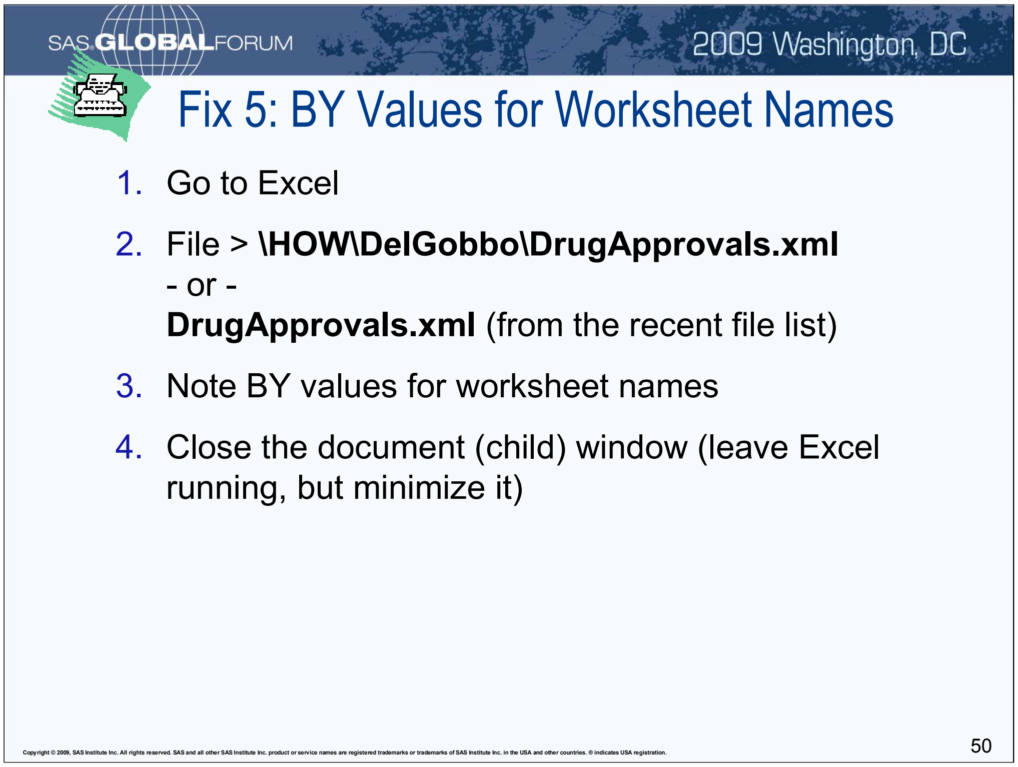
TO DO:

Line 24: Supply the tagset option `sheet_interval` with a value that specifies that new worksheets will be created for each new BY group value.

Line 24: Supply the tagset option `sheet_label` with a blank value.

Solution:

```
ods tagsets.ExcelXP options( ...
                   sheet_interval='bygroup'
                   sheet_label=' ');
```

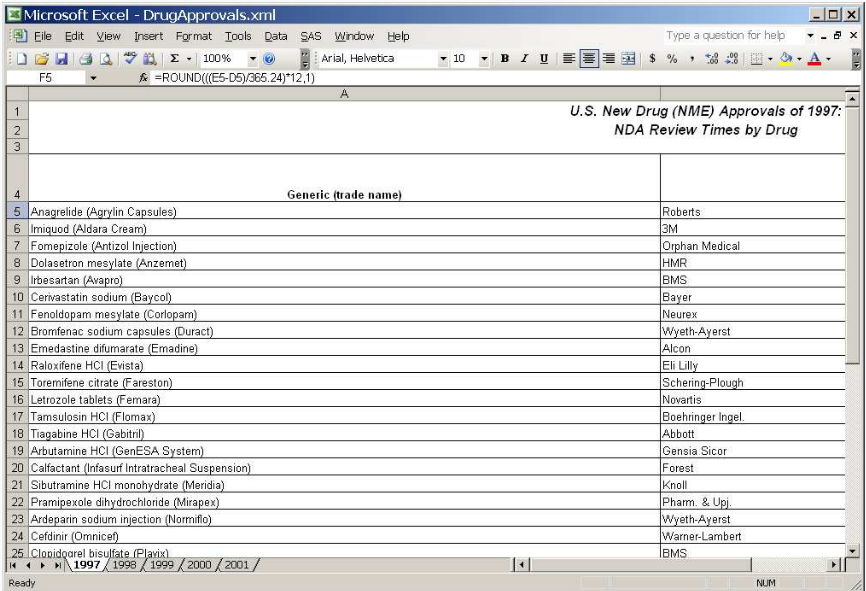


Fix 5: BY Values for Worksheet Names

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Note BY values for worksheet names
4. Close the document (child) window (leave Excel running, but minimize it)

Fix 6: Narrow Columns, Wrapped Text

Have:


Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Type a question for help
100%
Arial, Helvetica
10
Ready
NUM

F5 =ROUND(((E5-D5)/365.24)*12,1)

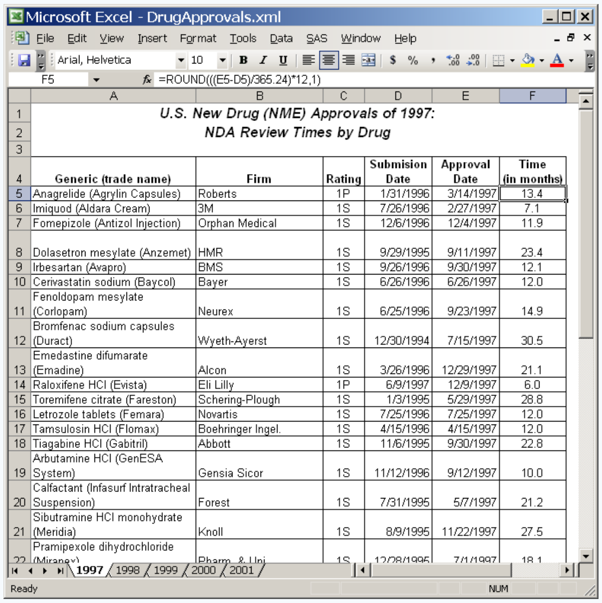
	A	
1	U.S. New Drug (NME) Approvals of 1997:	
2	NDA Review Times by Drug	
3		
4	**Generic (trade name)**	
5	Anagrelide (Agrylin Capsules)	Roberts
6	Imiquod (Aldara Cream)	3M
7	Fomepizole (Antizol Injection)	Orphan Medical
8	Dolasetron mesylate (Anzemet)	HMR
9	Irbesartan (Avapro)	BMS
10	Cerivastatin sodium (Baycol)	Bayer
11	Fenoldopam mesylate (Corlopam)	Neurex
12	Bromfenac sodium capsules (Duract)	Wyeth-Ayerst
13	Emedastine difumarate (Emadine)	Alcon
14	Raloxifene HCl (Evista)	Eli Lilly
15	Toremifene citrate (Fareston)	Schering-Plough
16	Letrozole tablets (Femara)	Novartis
17	Tamsulosin HCl (Flomax)	Boehringer Ingel.
18	Tiagabine HCl (Gabitril)	Abbott
19	Arbutamine HCl (GenESA System)	Gensia Sicor
20	Calfactant (Infasurf Intratracheal Suspension)	Forest
21	Sibutramine HCl monohydrate (Meridia)	Knoll
22	Pramipexole dihydrochloride (Mirapex)	Pharm. & Upj.
23	Ardeparin sodium injection (Normiflo)	Wyeth-Ayerst
24	Cefdinir (Omnicef)	Warner-Lambert
25	Clopidogrel bisulfate (Plavix)	BMS

1997 / 1998 / 1999 / 2000 / 2001

Wide columns.

Fix 6: Narrow Columns, Wrapped Text

Want:

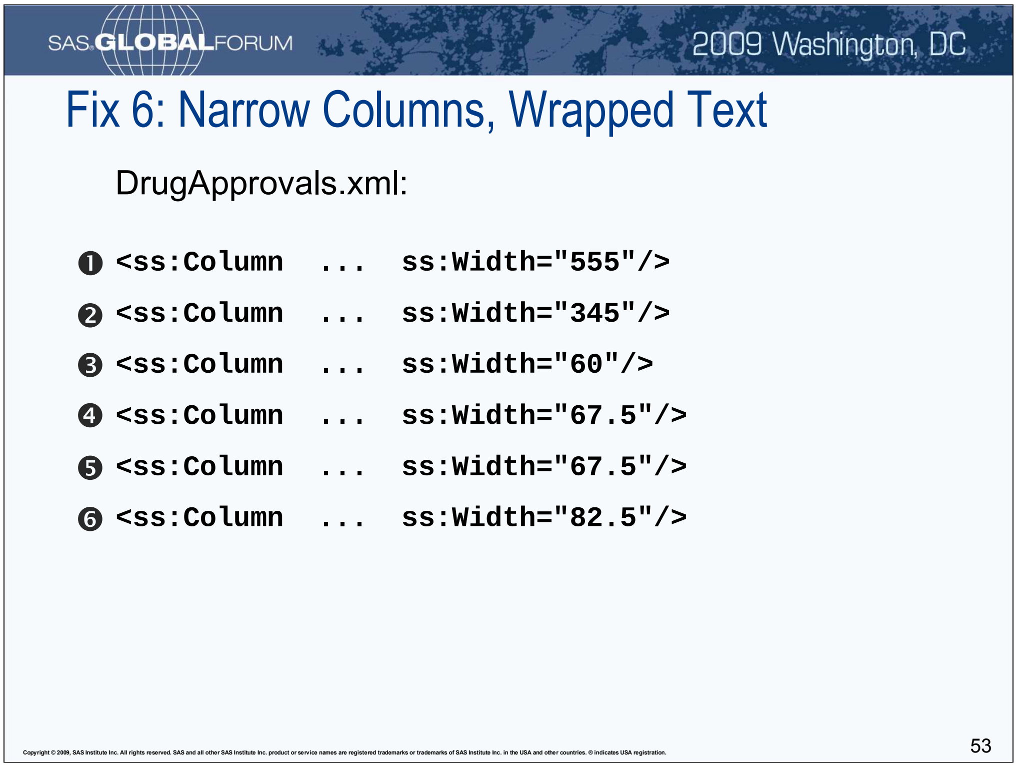

Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Arial, Helvetica
10
1997
1998
1999
2000
2001
Ready
NUM

```
F5    =ROUND(((E5-D5)/365.24)*12,1)
```

U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug

	Generic (trade name)	Firm	Rating	Submision Date	Approval Date	Time (in months)
5	Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
6	Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
7	Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
8	Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4
9	Irbesartan (Avapro)	BMS	1S	9/26/1996	9/30/1997	12.1
10	Cerivastatin sodium (Baycol)	Bayer	1S	6/26/1996	6/26/1997	12.0
11	Fenoldopam mesylate (Corlopam)	Neurex	1S	6/25/1996	9/23/1997	14.9
12	Bromfenac sodium capsules (Duract)	Wyeth-Ayerst	1S	12/30/1994	7/15/1997	30.5
13	Emedastine difumarate (Emadine)	Alcon	1S	3/26/1996	12/29/1997	21.1
14	Raloxifene HCl (Evista)	Eli Lilly	1P	6/9/1997	12/9/1997	6.0
15	Toremifene citrate (Fareston)	Schering-Plough	1S	1/3/1995	5/29/1997	28.8
16	Letrozole tablets (Femara)	Novartis	1S	7/25/1996	7/25/1997	12.0
17	Tamsulosin HCl (Flomax)	Boehringer Ingel.	1S	4/15/1996	4/15/1997	12.0
18	Tiagabine HCl (Gabitril)	Abbott	1S	11/6/1995	9/30/1997	22.8
19	Arbutamine HCl (GenESA System)	Gensia Sicor	1S	11/12/1996	9/12/1997	10.0
20	Calfactant (Infasurf Intratracheal Suspension)	Forest	1S	7/31/1995	5/7/1997	21.2
21	Sibutramine HCl monohydrate (Meridia)	Knoll	1S	8/9/1995	11/22/1997	27.5
22	Pramipexole dihydrochloride (Mirapex)	Pharm. & Upj	1S	12/28/1995	7/1/1997	18.1

Narrow columns with wrapped text.



Fix 6: Narrow Columns, Wrapped Text

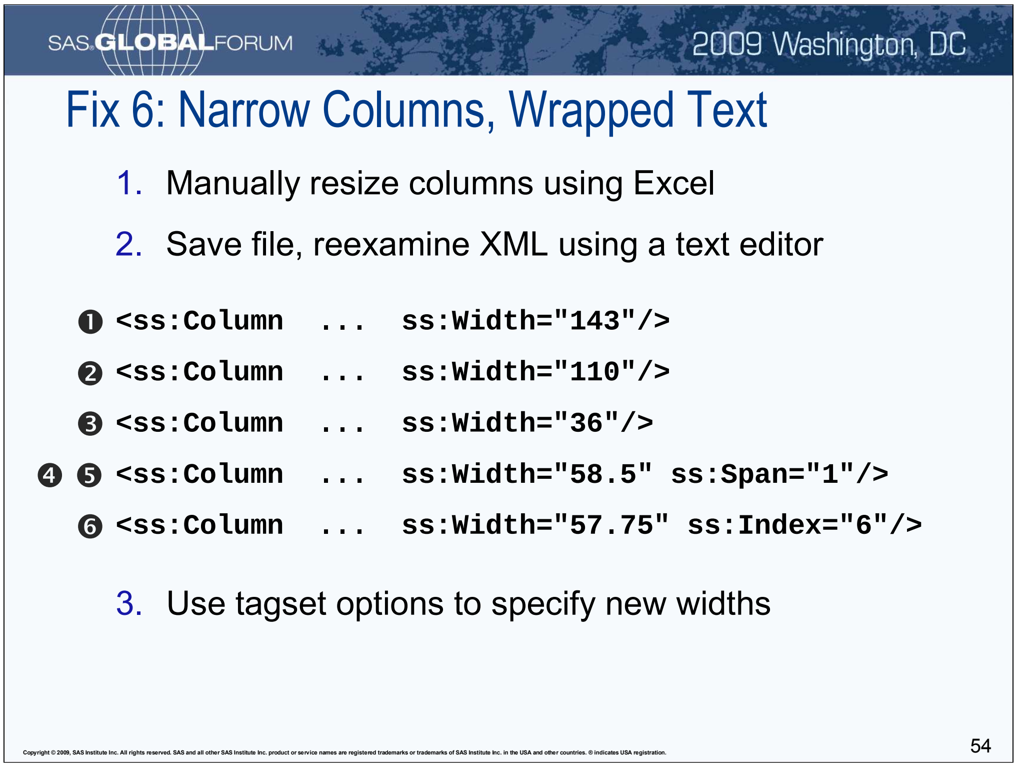
DrugApprovals.xml:

```
❶ <ss:Column  ...  ss:Width="555"/>
❷ <ss:Column  ...  ss:Width="345"/>
❸ <ss:Column  ...  ss:Width="60"/>
❹ <ss:Column  ...  ss:Width="67.5"/>
❺ <ss:Column  ...  ss:Width="67.5"/>
❻ <ss:Column  ...  ss:Width="82.5"/>
```

If you view the "DrugApprovals.xml" file using a text editor, you will see that the column definitions for all 5 worksheets look similar to what is shown here. The values of the "Width" attribute are specified in points.

Notice that the width values for the first two columns are very large. This happens because the algorithm that the tagset uses to compute the width takes into account the length of the variables, which for the first two columns are 74 and 46, respectively. Some of the other columns are also wider than they need to be to display the data.

You might instinctively try to narrow the columns by using a smaller value for the lengths of the character variables, but that can result in the truncation of data, and will have no effect on the width of the numeric columns.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 6: Narrow Columns, Wrapped Text

1. Manually resize columns using Excel
2. Save file, reexamine XML using a text editor

```
  ❶ <ss:Column  ...  ss:Width="143"/>
  ❷ <ss:Column  ...  ss:Width="110"/>
  ❸ <ss:Column  ...  ss:Width="36"/>
❹ ❺ <ss:Column  ...  ss:Width="58.5" ss:Span="1"/>
  ❻ <ss:Column  ...  ss:Width="57.75" ss:Index="6"/>
```

3. Use tagset options to specify new widths

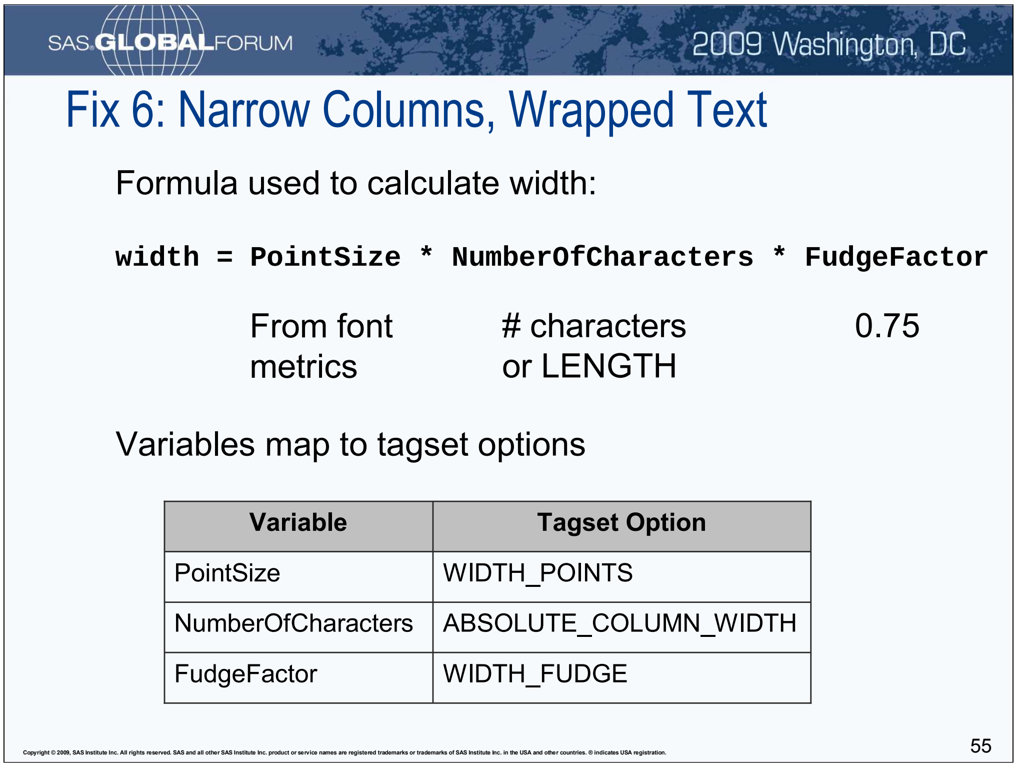
A better solution to finding optimal column widths is to perform the following steps for each worksheet:

1. Manually resize each column using Excel.
2. Save the XML file and reexamine the values of each "Width" attribute.
3. Use tagset options to specify the new widths.

When you manually resize the columns so that they are an acceptable width, and then you reexamine the XML file, you will see something like what is shown here.

Notice our worksheets contain 6 columns, but the XML created by saving the file using Excel has only 5 "Column" tags. Excel collapsed the 4th and 5th column tags into a single tag because the columns are the same width (58.5 points).

These are the values we want the tagset to use for the column widths.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 6: Narrow Columns, Wrapped Text

Formula used to calculate width:

```
width = PointSize * NumberOfCharacters * FudgeFactor
```

PointSize	NumberOfCharacters	FudgeFactor
From font metrics	# characters or LENGTH	0.75

Variables map to tagset options

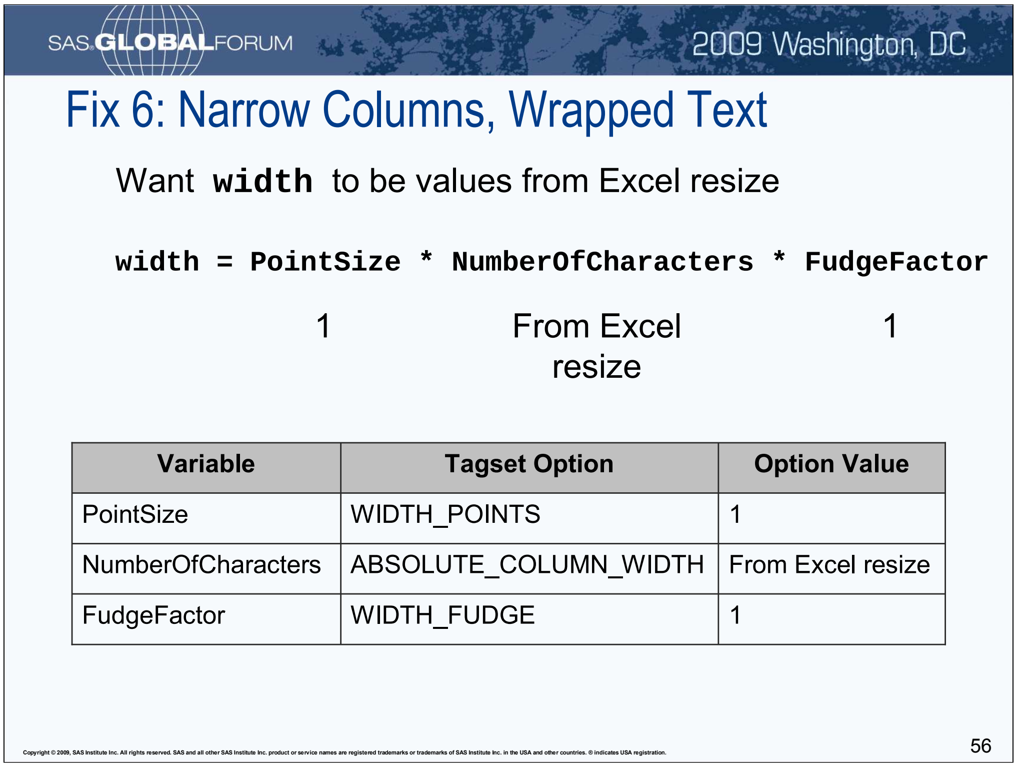
Variable	Tagset Option
PointSize	WIDTH_POINTS
NumberOfCharacters	ABSOLUTE_COLUMN_WIDTH
FudgeFactor	WIDTH_FUDGE

The tagset uses this formula to compute the approximate column width.

The value of "PointSize" is computed based on the metrics of the font. For numeric columns, "NumberOfCharacters" is the largest number of characters in any cell of the column, including the column heading. For character columns, "NumberOfCharacters" is defined as whichever is the larger: the length of the character variable, or the largest number of characters in a cell in the column. "FudgeFactor" has a default value of 0.75, and can be used to make small adjustments to the column width.

Values for each of the variables in the above equation can be specified using tagset options. The value for "PointSize" is specified using the WIDTH_POINTS option, "NumberOfCharacters" is specified using the ABSOLUTE_COLUMN_WIDTH option, and the WIDTH_FUDGE option is used to specify the value of "FudgeFactor".

Thus we have complete control of the value used for the column width.



Fix 6: Narrow Columns, Wrapped Text

Want **`width`** to be values from Excel resize

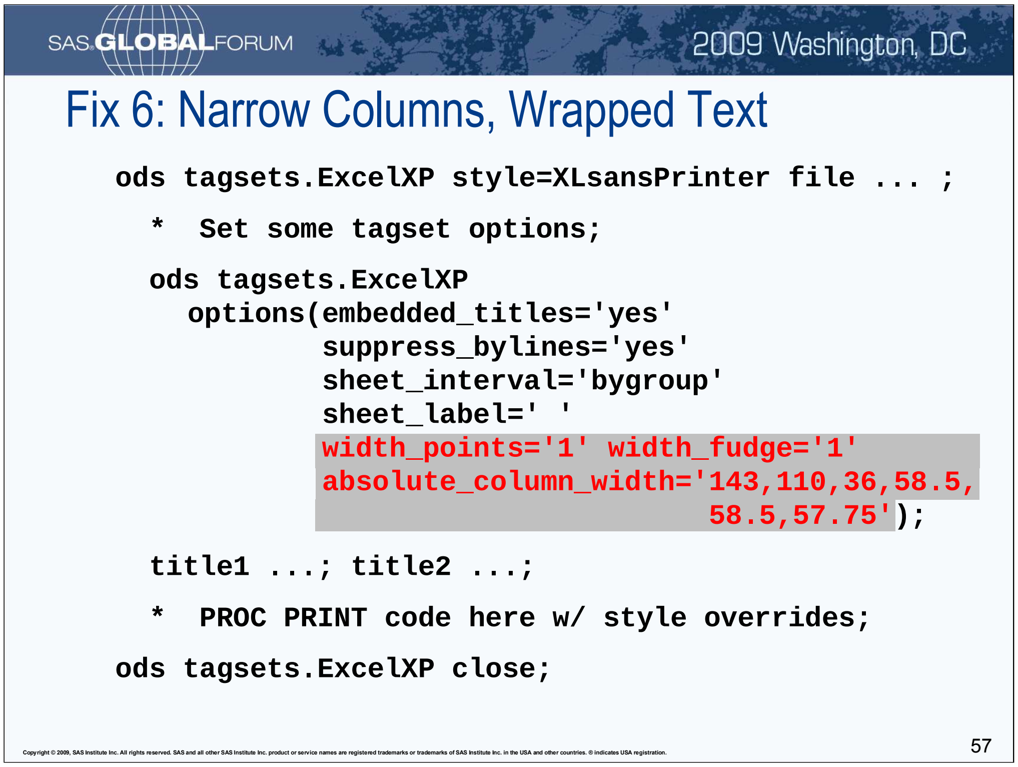
```
width = PointSize * NumberOfCharacters * FudgeFactor
```

1 From Excel resize 1

Variable	Tagset Option	Option Value
PointSize	WIDTH_POINTS	1
NumberOfCharacters	ABSOLUTE_COLUMN_WIDTH	From Excel resize
FudgeFactor	WIDTH_FUDGE	1

We can use the ABSOLUTE_COLUMN_WIDTH option to pass the new column widths to the tagset, but the final width will be recalculated based on the formula above.

However, setting the values of "PointSize" and "FudgeFactor" to 1 will effectively pass the values specified in ABSOLUTE_COLUMN_WIDTH directly into the XML file.



SAS GLOBAL FORUM

2009 Washington, DC

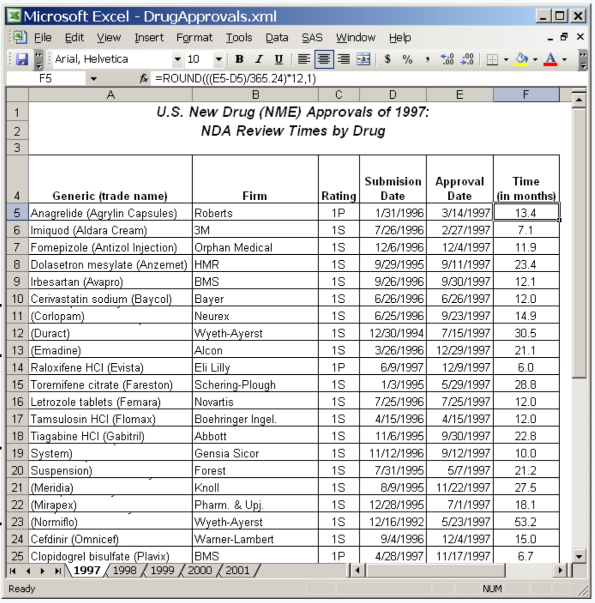
Fix 6: Narrow Columns, Wrapped Text

```
ods tagsets.ExcelXP style=XLsansPrinter file ... ;
  *  Set some tagset options;
  ods tagsets.ExcelXP
    options(embedded_titles='yes'
            suppress_bylines='yes'
            sheet_interval='bygroup'
            sheet_label=' '
            width_points='1' width_fudge='1'
            absolute_column_width='143,110,36,58.5,
                                   58.5,57.75');
  title1 ...; title2 ...;
  *  PROC PRINT code here w/ style overrides;
ods tagsets.ExcelXP close;
```

The 6 values specified for the ABSOLUTE_COLUMN_WIDTH option were obtained from the XML file, after the columns were resized using Excel.

These values are passed directly into the XML file because the value of the WIDTH_POINTS and WIDTH_FUDGE options are set to 1.

Fix 6: Narrow Columns, Wrapped Text


Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Arial, Helvetica
10
F5
1997
1998
1999
2000
2001
Ready
NUM

=ROUND(((E5-D5)/365.24)*12,1)

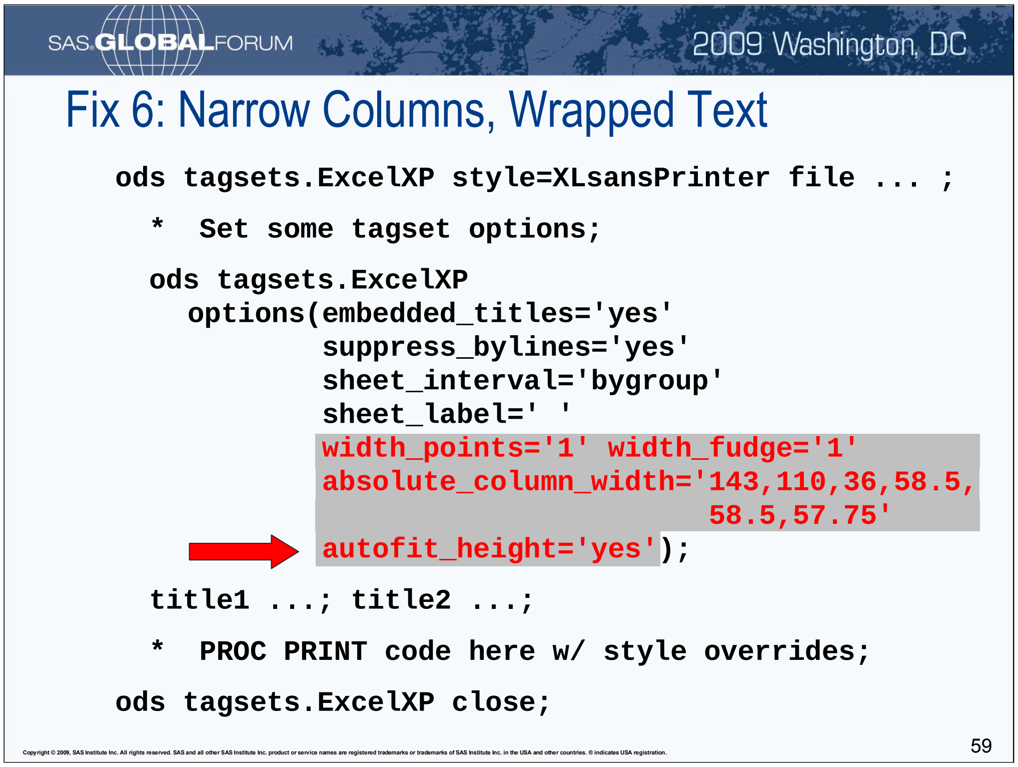
U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug

	A	B	C	D	E	F
4	**Generic (trade name)**	**Firm**	**Rating**	**Submision Date**	**Approval Date**	**Time (in months)**
5	Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
6	Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
7	Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
8	Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4
9	Irbesartan (Avapro)	BMS	1S	9/26/1996	9/30/1997	12.1
10	Cerivastatin sodium (Baycol)	Bayer	1S	6/26/1996	6/26/1997	12.0
11	(Corlopam)	Neurex	1S	6/25/1996	9/23/1997	14.9
12	(Duract)	Wyeth-Ayerst	1S	12/30/1994	7/15/1997	30.5
13	(Emadine)	Alcon	1S	3/26/1996	12/29/1997	21.1
14	Raloxifene HCl (Evista)	Eli Lilly	1P	6/9/1997	12/9/1997	6.0
15	Toremifene citrate (Fareston)	Schering-Plough	1S	1/3/1995	5/29/1997	28.8
16	Letrozole tablets (Femara)	Novartis	1S	7/25/1996	7/25/1997	12.0
17	Tamsulosin HCl (Flomax)	Boehringer Ingel.	1S	4/15/1996	4/15/1997	12.0
18	Tiagabine HCl (Gabitril)	Abbott	1S	11/6/1995	9/30/1997	22.8
19	System)	Gensia Sicor	1S	11/12/1996	9/12/1997	10.0
20	Suspension)	Forest	1S	7/31/1995	5/7/1997	21.2
21	(Meridia)	Knoll	1S	8/9/1995	11/22/1997	27.5
22	(Mirapex)	Pharm. & Upj.	1S	12/28/1995	7/1/1997	18.1
23	(Normiflo)	Wyeth-Ayerst	1S	12/16/1992	5/23/1997	53.2
24	Cefdinir (Omnicef)	Warner-Lambert	1S	9/4/1996	12/4/1997	15.0
25	Clopidogrel bisulfate (Plavix)	BMS	1P	4/28/1997	11/17/1997	6.7

Partial display of data

Submitting the most recent code results in columns that have appropriate widths, but some of the rows are not as tall as they need to be.

For example, rows 11-13 and 19-23 are not tall enough, resulting in only a partial display of the drug names.

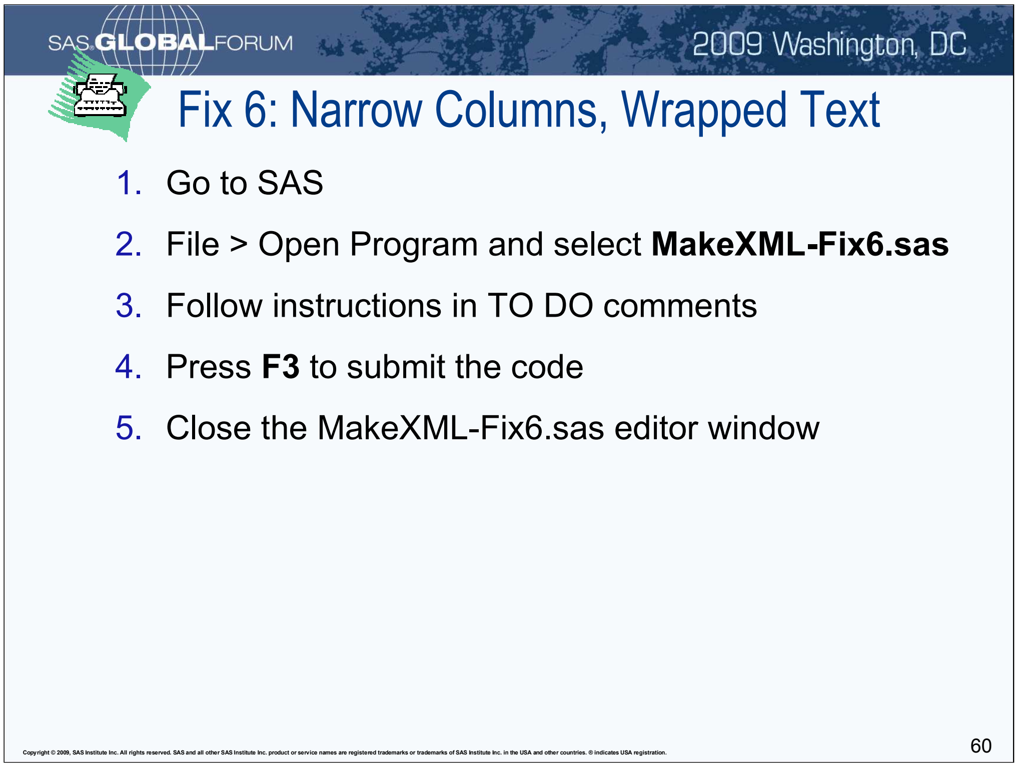


SAS GLOBAL FORUM 2009 Washington, DC

Fix 6: Narrow Columns, Wrapped Text

```
ods tagsets.ExcelXP style=XLsansPrinter file ... ;
  *  Set some tagset options;
  ods tagsets.ExcelXP
    options(embedded_titles='yes'
            suppress_bylines='yes'
            sheet_interval='bygroup'
            sheet_label=' '
            width_points='1' width_fudge='1'
            absolute_column_width='143,110,36,58.5,
                                   58.5,57.75'
            autofit_height='yes');
  title1 ...; title2 ...;
  *  PROC PRINT code here w/ style overrides;
ods tagsets.ExcelXP close;
```

To make the row height adjust to its content, we use the AUTOFIT_HEIGHT tagset option.



SAS GLOBAL FORUM 2009 Washington, DC

Fix 6: Narrow Columns, Wrapped Text

1. Go to SAS
2. File > Open Program and select **MakeXML-Fix6.sas**
3. Follow instructions in TO DO comments
4. Press **F3** to submit the code
5. Close the MakeXML-Fix6.sas editor window

Copyright © 2009, SAS Institute Inc. All rights reserved. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

60

TO DO:

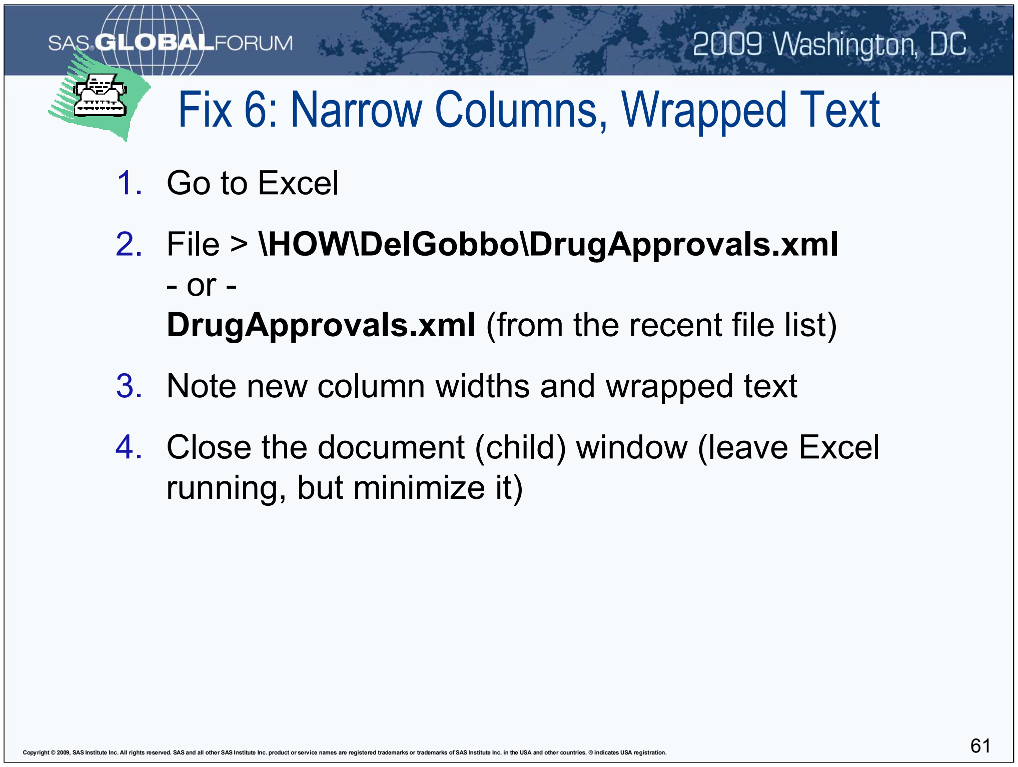
Line 28: Supply the tagset options to set the point size and "fudge factor" to a value of 1.

Line 29: Note the column widths specified using the tagset option `absolute_column_width`.

Line 30: Supply the tagset option to cause the row heights to be "autofit".

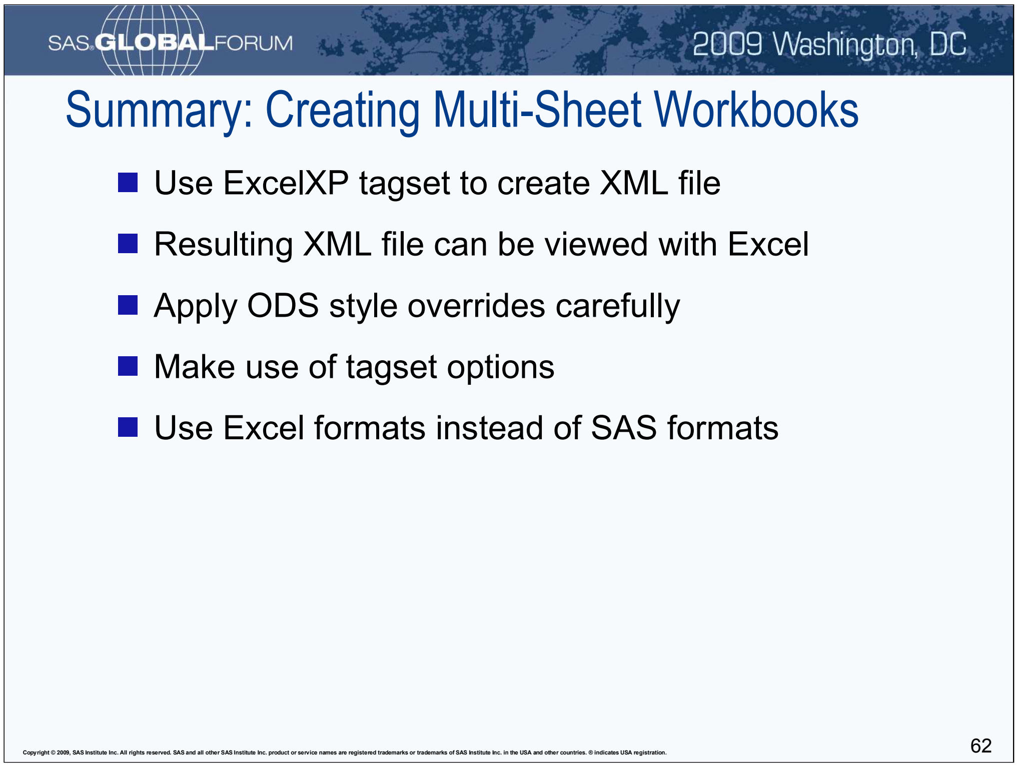
Solution:

```
ods tagsets.ExcelXP options( ...
  width_points='1' width_fudge='1'
  absolute_column_width='143,110,36,58.5,58.5,57.75'
  autofit_height='yes');
```



Fix 6: Narrow Columns, Wrapped Text

1. Go to Excel
2. File > **\HOW\DelGobbo\DrugApprovals.xml**
   - or -
   DrugApprovals.xml (from the recent file list)
3. Note new column widths and wrapped text
4. Close the document (child) window (leave Excel running, but minimize it)



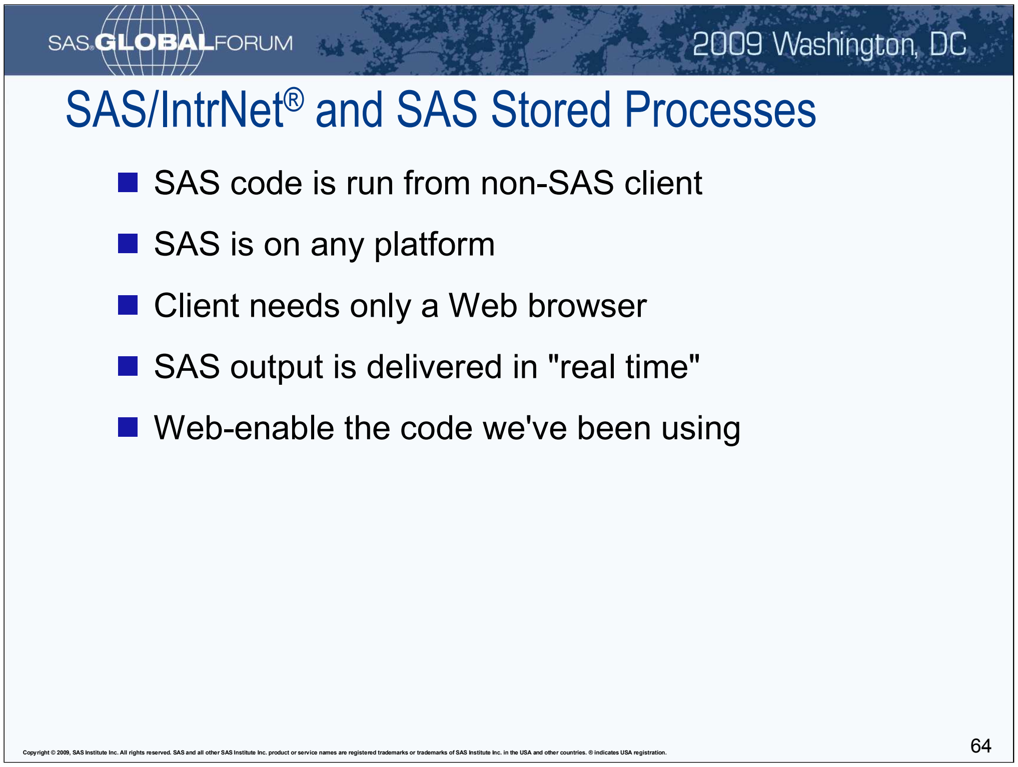
Summary: Creating Multi-Sheet Workbooks

- Use ExcelXP tagset to create XML file
- Resulting XML file can be viewed with Excel
- Apply ODS style overrides carefully
- Make use of tagset options
- Use Excel formats instead of SAS formats

Using SAS/IntrNet®

and

SAS Stored Processes



SAS/IntrNet® and SAS Stored Processes

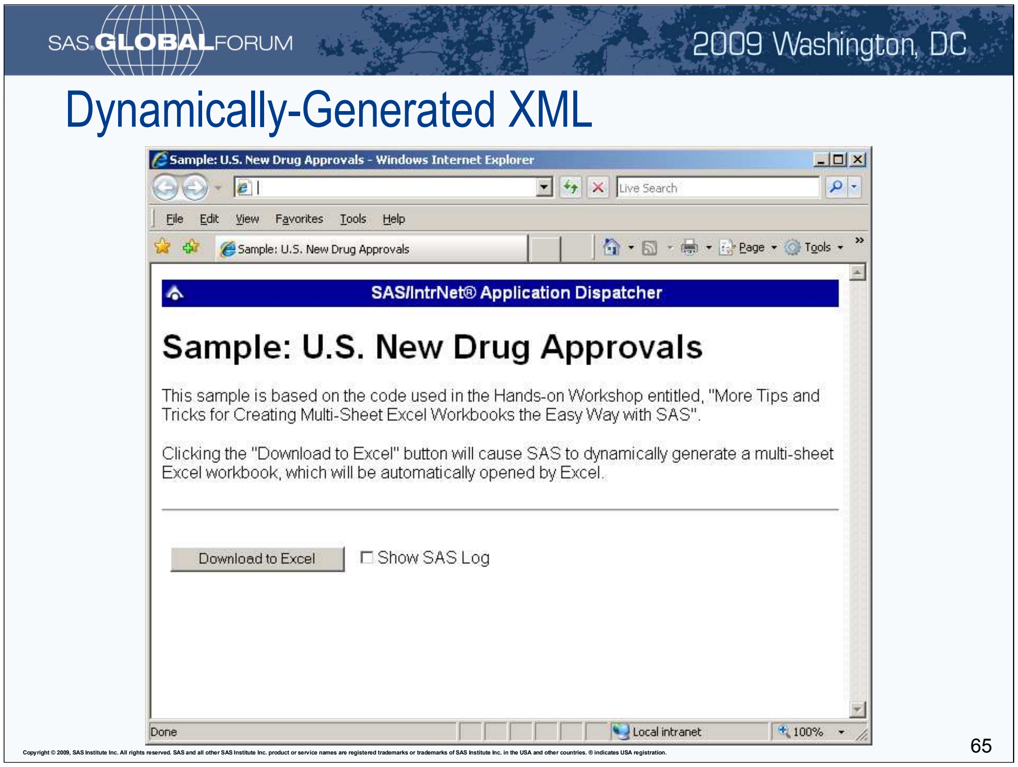
- SAS code is run from non-SAS client
- SAS is on any platform
- Client needs only a Web browser
- SAS output is delivered in "real time"
- Web-enable the code we've been using

The purpose of the SAS/IntrNet Application Dispatcher or SAS Stored Processes are to allow you to execute SAS programs from a machine that does not have SAS installed. The machine *may* have SAS installed, but that is not required.

A typical client-server model is followed. The SAS server can reside on any hardware platform (Windows, UNIX, z/OS, etc.) and is standing by, waiting to execute a SAS program. The most common client is a Web browser, again, running on any platform.

When the "OK" button of the Web page is clicked, input parameters, if any, are sent to the SAS server. Your SAS code executes, and the output is delivered in "real time" to the Web browser.

The following slides illustrate this process, using a "Web-enabled" version of the SAS code we have been working with.


SAS GLOBAL FORUM
2009 Washington, DC
Sample: U.S. New Drug Approvals - Windows Internet Explorer
Live Search
File Edit View Favorites Tools Help
Sample: U.S. New Drug Approvals
Page
Tools
Done
Local intranet
100%

Dynamically-Generated XML

SAS/IntrNet® Application Dispatcher

Sample: U.S. New Drug Approvals

This sample is based on the code used in the Hands-on Workshop entitled, "More Tips and Tricks for Creating Multi-Sheet Excel Workbooks the Easy Way with SAS".

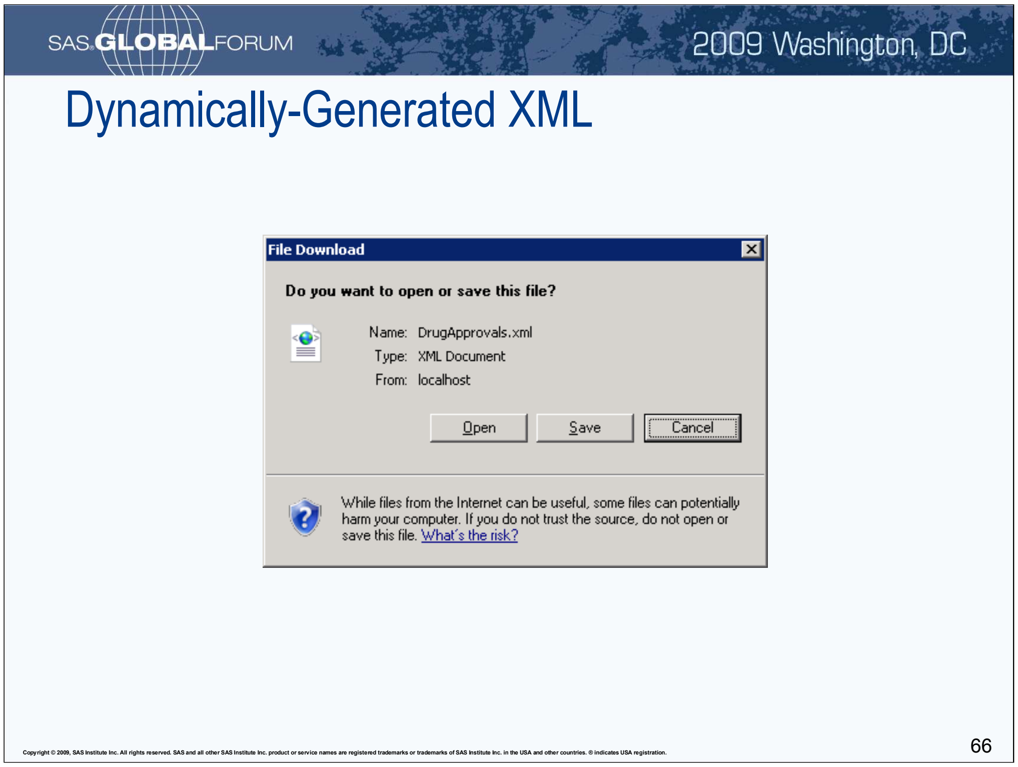
Clicking the "Download to Excel" button will cause SAS to dynamically generate a multi-sheet Excel workbook, which will be automatically opened by Excel.

Download to Excel ☐ Show SAS Log

Here is a simple Web page that is used to execute SAS code stored on a server using the SAS/IntrNet product.

The code that is executed is substantially similar to the final versions of the code that we used to generate XML file, with a few changes to "Web-enable" it.

Clicking "Download to Excel" causes the SAS program to execute on the server.



SAS GLOBAL FORUM 2009 Washington, DC

Dynamically-Generated XML

Do you want to open or save this file?

Name: DrugApprovals.xml
Type: XML Document
From: localhost

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

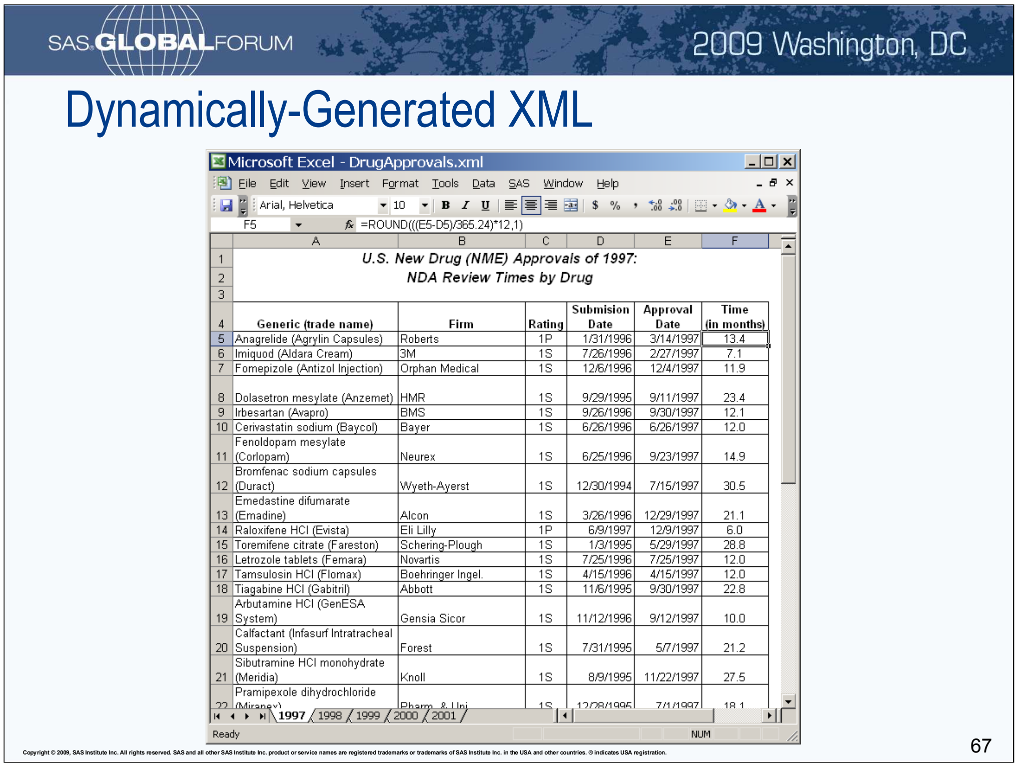
Once the SAS program executes, the results are sent back to the Web browser.

Note that you are presented with a File Download dialog, instead of the results being displayed in the Web browser. Clicking "Open" causes the SAS output to be displayed in Excel, provided that Excel is installed on the client machine.

This SAS/IntrNet-specific code, that must be specified before any ODS statements, is used to cause the SAS output to be displayed in Excel:

```
%let RV =%sysfunc(appsrv_header(Content-type,application/vnd.ms-excel));
```

This code will also work with SAS Stored Processes.


Microsoft Excel - DrugApprovals.xml
File Edit View Insert Format Tools Data SAS Window Help
Arial, Helvetica
10
F5
=ROUND(((E5-D5)/365.24)*12,1)
1997
1998
1999
2000
2001
Ready
NUM

Dynamically-Generated XML

U.S. New Drug (NME) Approvals of 1997:
NDA Review Times by Drug

Generic (trade name)	Firm	Rating	Submision Date	Approval Date	Time (in months)
Anagrelide (Agrylin Capsules)	Roberts	1P	1/31/1996	3/14/1997	13.4
Imiquod (Aldara Cream)	3M	1S	7/26/1996	2/27/1997	7.1
Fomepizole (Antizol Injection)	Orphan Medical	1S	12/6/1996	12/4/1997	11.9
Dolasetron mesylate (Anzemet)	HMR	1S	9/29/1995	9/11/1997	23.4
Irbesartan (Avapro)	BMS	1S	9/26/1996	9/30/1997	12.1
Cerivastatin sodium (Baycol)	Bayer	1S	6/26/1996	6/26/1997	12.0
Fenoldopam mesylate (Corlopam)	Neurex	1S	6/25/1996	9/23/1997	14.9
Bromfenac sodium capsules (Duract)	Wyeth-Ayerst	1S	12/30/1994	7/15/1997	30.5
Emedastine difumarate (Emadine)	Alcon	1S	3/26/1996	12/29/1997	21.1
Raloxifene HCl (Evista)	Eli Lilly	1P	6/9/1997	12/9/1997	6.0
Toremifene citrate (Fareston)	Schering-Plough	1S	1/3/1995	5/29/1997	28.8
Letrozole tablets (Femara)	Novartis	1S	7/25/1996	7/25/1997	12.0
Tamsulosin HCl (Flomax)	Boehringer Ingel.	1S	4/15/1996	4/15/1997	12.0
Tiagabine HCl (Gabitril)	Abbott	1S	11/6/1995	9/30/1997	22.8
Arbutamine HCl (GenESA System)	Gensia Sicor	1S	11/12/1996	9/12/1997	10.0
Calfactant (Infasurf Intratracheal Suspension)	Forest	1S	7/31/1995	5/7/1997	21.2
Sibutramine HCl monohydrate (Meridia)	Knoll	1S	8/9/1995	11/22/1997	27.5
Pramipexole dihydrochloride (Mirapex)	Pharm. & Upj	1S	12/28/1995	7/1/1997	18.1

Here is the SAS output, created in "real time", and delivered to the client. Note that Excel is used to view the SAS output, not the Web browser.

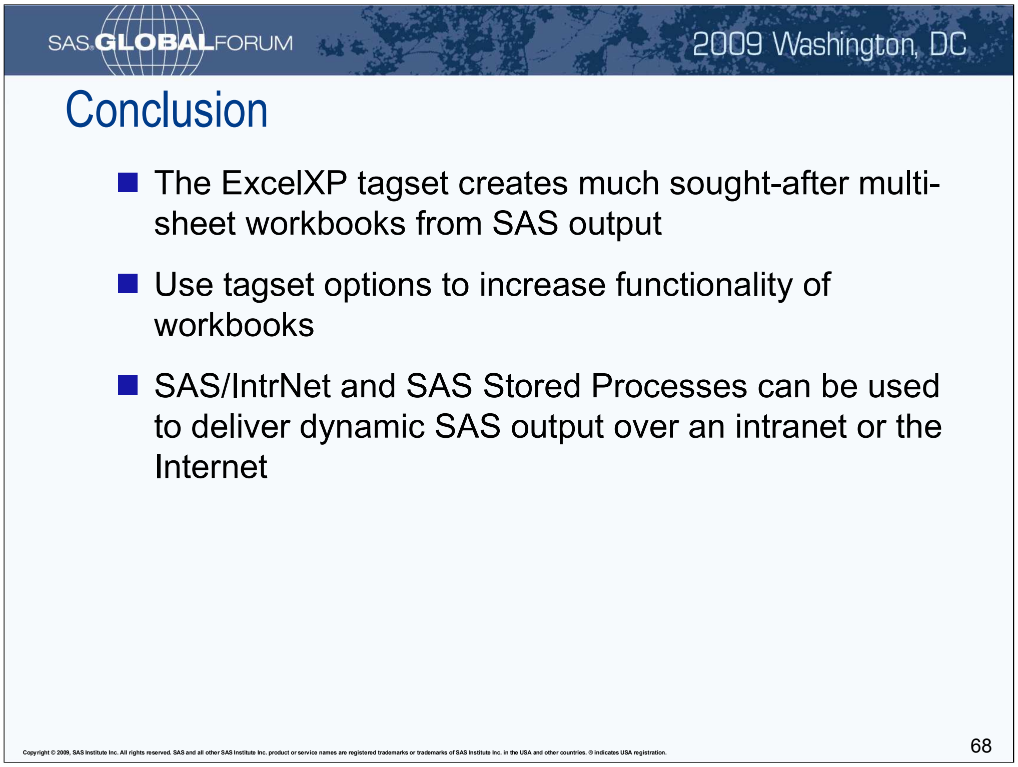
Refer to the accompanying paper for more details on this topic.

References:

SAS Institute Inc. "About Application Dispatcher". http://support.sas.com/onlinedoc/913/docMainpage.jsp?_topic=dispatch.hlp/main_contents.htm.

(Ignore wrapping in URL above)

SAS Institute Inc. "SAS Stored Processes". http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/index.html.



SAS GLOBAL FORUM 2009 Washington, DC

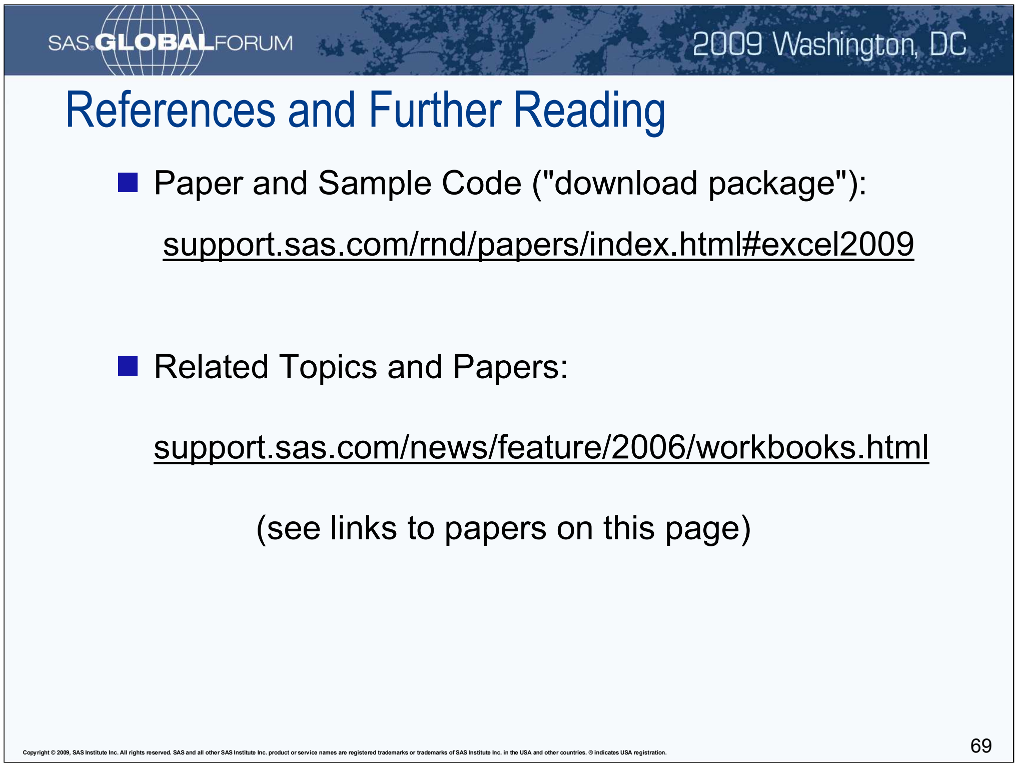
Conclusion

- The ExcelXP tagset creates much sought-after multi-sheet workbooks from SAS output
- Use tagset options to increase functionality of workbooks
- SAS/IntrNet and SAS Stored Processes can be used to deliver dynamic SAS output over an intranet or the Internet

Using ODS to generate specially-formatted XML output is an effective method of incorporating SAS output in Excel documents.

The SAS®9 ExcelXP ODS tagset provides an easy way to export your SAS data to Excel workbooks that contain multiple worksheets. By using ODS styles, style overrides, and a tagset that complies with the Microsoft XML Spreadsheet Specification, you can customize the output to achieve your design goals.

SAS Institute continues to work toward better Microsoft Office integration, and future releases of SAS software will provide even more robust means of using SAS content with Microsoft Office applications.



References and Further Reading

- Paper and Sample Code ("download package"):

  support.sas.com/rnd/papers/index.html#excel2009

- Related Topics and Papers:

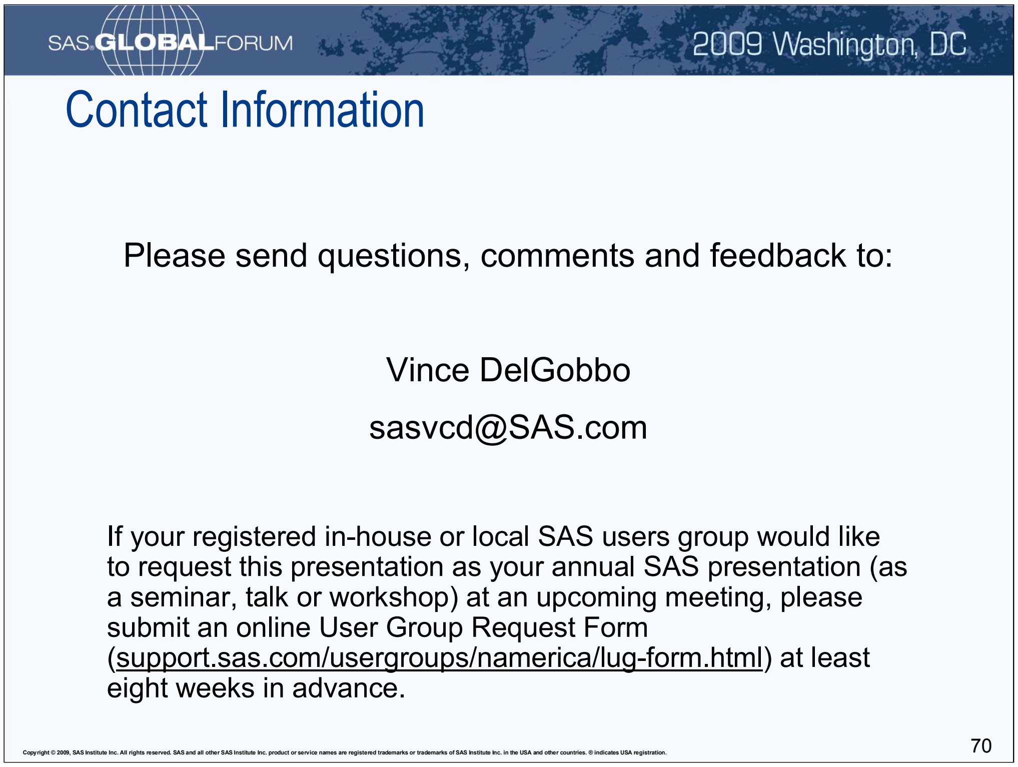
  support.sas.com/news/feature/2006/workbooks.html

  (see links to papers on this page)

The sample programs and data used in this workshop as well as a copy of the accompanying paper are available at the SAS Presents Web site. Go to http://support.sas.com/rnd/papers/index.html#excel2009 and find the entry "More Tips and Tricks for Creating Multi-Sheet Excel Workbooks the Easy Way with SAS".

IMPORTANT NOTE: Be sure to install a recent version of the ExcelXP tagset on your system. See the accompanying paper for details and instructions.

Review the "Installation" and "Usage" sections of the file named "ReadMe.txt" in the ZIP archive for important information on using the sample files.



SAS GLOBAL FORUM 2009 Washington, DC

Contact Information

Please send questions, comments and feedback to:

Vince DelGobbo

sasvcd@SAS.com

If your registered in-house or local SAS users group would like to request this presentation as your annual SAS presentation (as a seminar, talk or workshop) at an upcoming meeting, please submit an online User Group Request Form (support.sas.com/usergroups/namerica/lug-form.html) at least eight weeks in advance.

About the author:

Vince DelGobbo is a Senior Systems Developer in the Web Tools group at SAS. This group is responsible for developing the SAS/IntrNet Application Dispatcher and SAS Stored Processes. He is the developer of the HTML Formatting Tools and the SAS Design-Time Controls, and is developing other new Web- and server-based technologies, as well as integrating SAS output with Microsoft Office. He is also involved in the development of the ExcelXP ODS tagset. Vince has been a SAS Software user since 1982, and joined SAS in 1992.