Paper 318-2009

Have It Your Way: Rearrange and Replay Your Output with ODS DOCUMENT

Cynthia L. Zender, SAS Institute Inc., Cary, NC

ABSTRACT

Is everything about your SAS® output the way you want, except for the final order of the report objects? Do you want your output organized by year or by city, instead of by SAS procedure, without using SAS macro programs? Do you like the order that SAS produces, but you just want to change the folder hierarchy?

This paper illustrates how to capture your output and save it in an ODS document store. Then, you can create custom folders and a custom folder hierarchy using ODS DOCUMENT and PROC DOCUMENT to rearrange and replay your output. Along the way, you learn how to add custom titles and notes to your output; delete page breaks; label and rename folders; and use PROC DOCUMENT syntax to automate the creation of reports in the format you want. The ODS DOCUMENT window is demonstrated, along with PROC DOCUMENT. Concrete examples are used to illustrate every major task. The final report includes a Table of Contents that showcases the new report structure. Please note that the *Proceedings of the SAS Global Forum 2009 Conference* did not contain a copy of this paper because of the voluminous screen shots needed to show you ODS DOCUMENT with concrete examples, and the code needed to illustrate PROC DOCUMENT concepts.

INTRODUCTION

ODS DOCUMENT is an ODS destination. But, unlike "report" destinations, such as HTML, RTF, and PDF that create output files that are viewed and rendered with a third-party viewer, ODS DOCUMENT is not a SAS report destination. It is more like the ODS OUTPUT destination—in other words, a "data" destination. The ODS DOCUMENT destination holds output objects that are based on a SAS procedure or process. It holds output objects in their original creation structures, but you can rearrange the hierarchy and structure of the objects stored by ODS DOCUMENT. ODS documents are stored in a proprietary format (a document store) and can be viewed only with SAS software. The ODS DOCUMENT destination holds the output objects in such a way that they are able to be replayed or rerun to one of the report destinations. The ODS DOCUMENT destination is a SAS proprietary destination—to view or modify what's in the document store, you have to use either the ODS DOCUMENT window or PROC DOCUMENT.

The above description is well and good, and you might have a sense of what the ODS DOCUMENT destination is, but you're not convinced that you need it. How about this usage scenario? You have a job that runs for six hours that performs some complicated analysis with a bunch of program steps. The only copy of the report is in HTML on the company Web server. You can run the job only overnight. But, your boss comes in at 9:30 a.m. and says she needs a copy of the report in PDF, with page numbers, in landscape mode, for a noon meeting. You don't have six hours to rerun the analysis to ODS PDF. What do you do?

ODS DOCUMENT to the rescue! Fortunately, you froze copies of the analysis output objects in an ODS document store the last time you did the HTML report. All you have to do is submit a PROC DOCUMENT replay job to run the ODS DOCUMENT output objects to the ODS PDF destination. Lucky you! Your output, your way, and it didn't take another six hours to run the analysis.

Still not convinced? Here's another scenario. You have a PROC TABULATE step with BY-group processing, followed by a PROC UNIVARIATE step with BY-group processing. In the default folder structure created for the job, all of the BY groups for the PROC TABULATE step appear before the BY groups for the PROC UNIVARIATE step. Your boss is tired of flipping back and forth from one section of the output to another to compare the PROC TABULATE report with the EXTREMEOBS portion of the PROC UNIVARIATE report. Your boss requests that you organize the output report so that you have the PROC TABULATE report immediately followed by the EXTREMEOBS report for the same BY group. Oh, and while you're at it, she wants only the EXTREMEOBS portion from the PROC UNIVARIATE output. And, what's that? You don't have any budget for a SAS Macro Programming class. And, you have to produce this report in batch mode, every month, on demand, as needed.

Don't panic! If you save copies of your original output objects from PROC TABULATE and PROC UNIVARIATE in an ODS document store, you can rearrange the output objects in a hierarchical structure that is different from the original procedure-centric structure. Then, you can rerun the new ODS document to the destination of your choice. Even better, you can use PROC DOCUMENT statements in batch mode to create this output whenever you want—daily, weekly, monthly, quarterly. Although using SAS macro processing can make your program more generic and reusable, there's no macro programming required to get the output you want the way you want it.

Before we jump into the examples, let's look at some terminology and basic information about the ODS DOCUMENT destination.

OVERVIEW OF THE DOCUMENT DESTINATION

The ODS DOCUMENT destination enables you to freeze or store the output objects from your procedure or process in a special type of item store called a document store. An item store is a unique type of SAS structure. SAS templates and the SAS registry are kept in item stores. They are managed with different procedures. SAS templates are managed with PROC TEMPLATE, and the SAS registry with PROC REGISTRY. An ODS document store is managed using the interactive ODS DOCUMENT window or with PROC DOCUMENT.

You create a document store using the ODS DOCUMENT sandwich invocation:

```
ods listing close;

proc sort data=sashelp.prdsale out=prdsale;
  by Country;
run;

ods document name=work.prddoc(write);
proc tabulate data=prdsale;
  by Country;
  var predict;
  class prodtype;
  table prodtype all,
        predict*(min mean max);
run;

ods select ExtremeObs;
proc univariate data=prdsale;
  by Country;
  var actual;
run;
ods document close;
```

After the program runs, the SAS Results window is displayed. (See Figure 1.) Also displayed are the properties of the first output object in the results (for the first PROC TABULATE BY group).

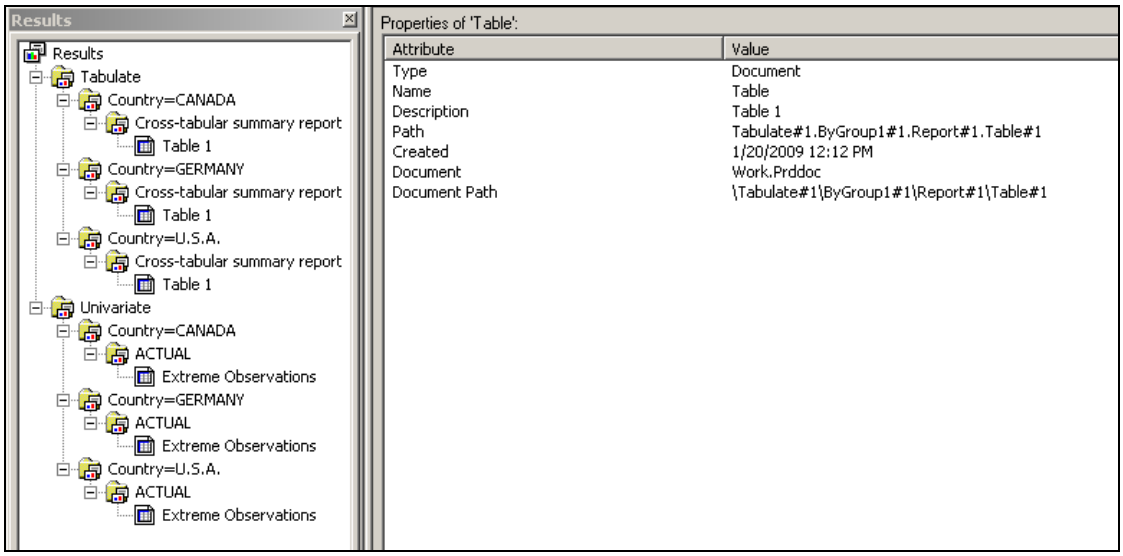


Figure 1. SAS Results Window Showing Document Icon and Properties with Type Attribute of Document

An ODS document store is not a SAS data set, as you can see by the document icon in Figure 1. This ODS document was written to the Work library. However, if it had been written to a permanent location such as **`c:\temp\output`**, then, when you looked in Windows Explorer after you created the document store, you would see

that the document store has a file extension of SAS7BITM (the same as a template store or the registry store). A screen shot of a document store in a permanent location is shown in Figure 2:

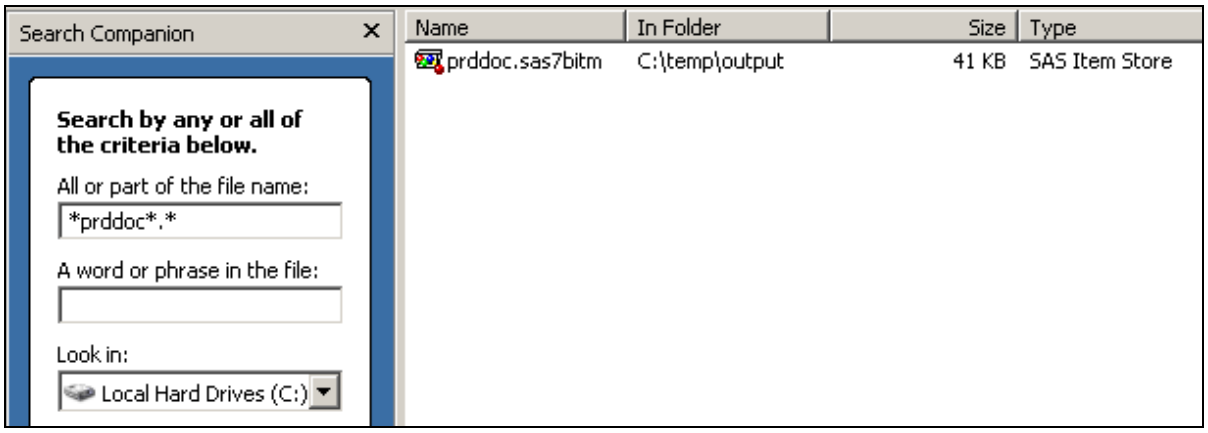


Figure 2. Permanent Document Store and Its File Extension

So, now you've created a SAS item store. What do you do with it? In this paper, we examine the ODS document store with PROC DOCUMENT, and then we use the ODS DOCUMENT window. We show both methods to modify the items in the document store and replay them.

EXPLORING YOUR ODS DOCUMENT STORE WITH PROC DOCUMENT

PROC DOCUMENT is the procedure designed for use with an ODS DOCUMENT store. The LIST statement of PROC DOCUMENT allows you to view the contents of the newly created ODS document. The REPLAY statement allows you to replay the document to any open destination. Remember that the NAME= option created the WORK.PRDDOC ODS document. You will find that you never use DATA= with PROC DOCUMENT. If you accidentally use DATA= in the code below, you would get an ERROR message that PROC DOCUMENT expects the NAME= option. That should help drive home the fact that an ODS document is not the same thing as a SAS data set. The NAME= option with the LIST statement will allow you to look at the object list and folder structure within the ODS document. Consider the following program.

```
proc document name=work.prddoc;
  list / levels=all;
run;
quit;
```

The LIST statement is one of the information statements. What you'll see in Figure 3 is that the LIST output is a text version of what was in the Results Window (as shown in Figure 1). Figure 3 has been annotated to show the correspondence between the Results Window and the LIST output for two of the output objects and their folders.

What you see in Figure 3 is that every folder icon in the Results Window corresponds to an item with a type of "Dir" in the LIST statement output. Every output object created by a procedure corresponds to an item with a type of "Table" in the LIST statement output.

The naming convention can seem a bit obscure with all the '#' and the numbers. You will note that the long name for the first output object (for the CANADA by group) for PROC TABULATE is:

```
\Tabulate#1\ByGroup1#1\Report#1\Table#1
```

This is the same as the "Document path" shown in Figure 1 for the properties of the Canada by group object. Every folder or directory name appears between the slashes (\) in the document pathname. While the Properties window is good for visually viewing an object name, the LIST statement output is good for cutting and pasting pathnames into other PROC DOCUMENT syntax (as we will see when copying and rearranging output objects).

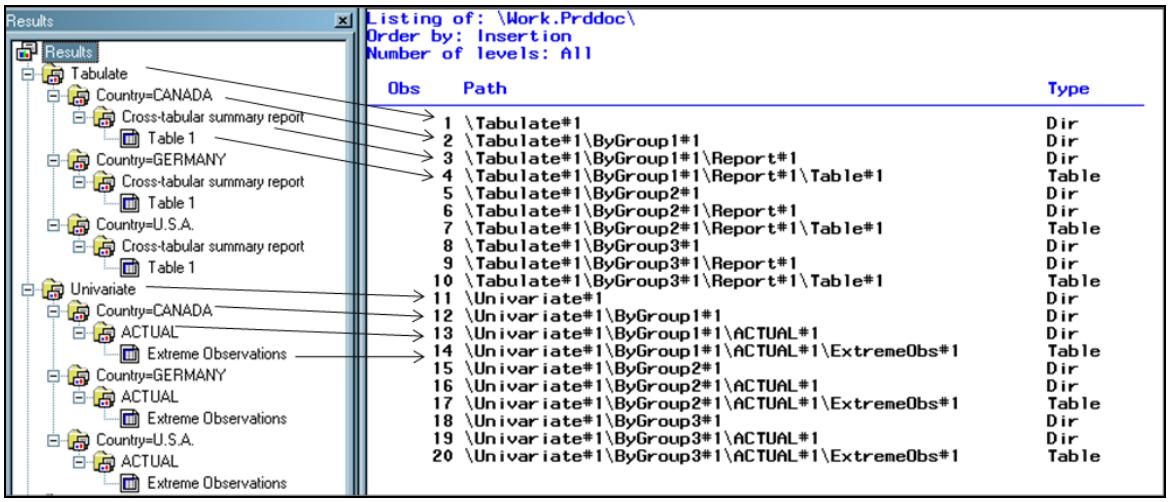
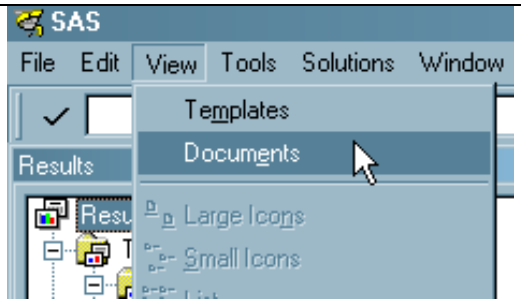
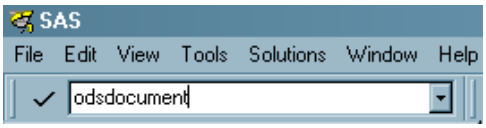


Figure 3: PROC DOCUMENT List Output Compared to Results Window

The Demo2 program that is available for download has some extra examples of how to use the LIST statement. In addition to what is contained there, the documentation for ODS DOCUMENT has some excellent examples of using WHERE expressions with a group of PROC DOCUMENT statements (including the LIST statement), as discussed in the documentation topic, "Using WHERE Expressions with the DOCUMENT Procedure". This PROC DOCUMENT method of getting property information is a great tool; however, an even easier way to interact with your ODS Document is to use the interactive Document window.

EXPLORING YOUR ODS DOCUMENT STORE WITH THE DOCUMENT WINDOW

The next series of figures illustrates exploring WORK.PRDDOC with the Document window.

Task and Notes	Figure
To open the Document Window, use one of these two methods: Select **View → Documents** from the menu OR type **'odsdocument'** or **'odsdoc'** in the command area.	 **OR**  **Figure 4**

To expand the current document, highlight the ODS Document of interest, right-click, and select **Expand All**.

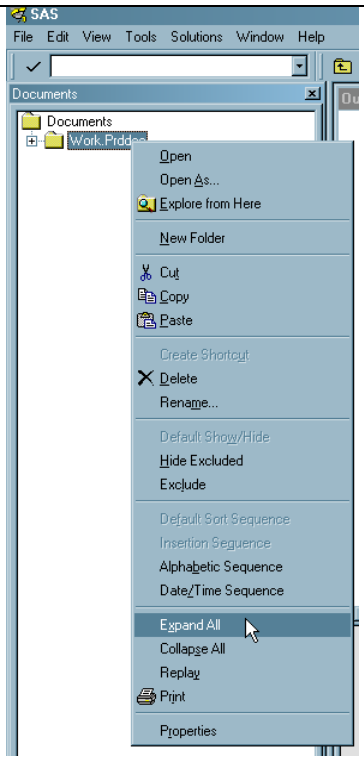


Figure 5

This view of the expanded Document Window shows that it is docked beside the Results Window and the Explorer Window.

To undock any of these windows, click on the tab at the bottom of the window and the choice "Docked" should be checked. To unselect or undock the window, click on the word "Docked" to uncheck the window.

If you undock the Results window and the Document window in this fashion, you can rearrange the two windows to be side by side, as shown in Figure 7.

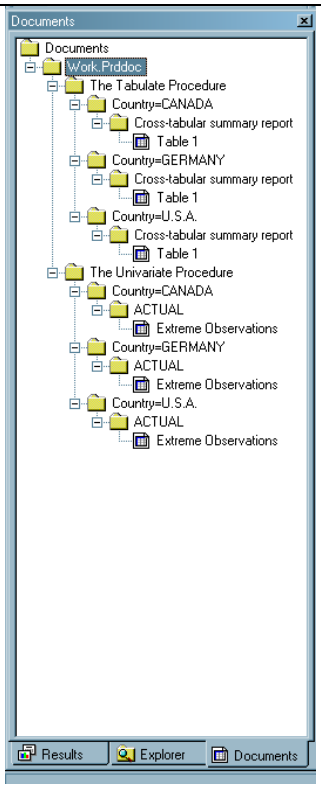


Figure 6

If you undock the Document Window and the Results Window and place them side by side, you can see how the folder structure of WORK.PRDDOC (the ODS Document) is the same as the structure of the results hierarchy tree in the Results window.

To view the properties of an item in the Window, right-click on the item and select **Properties**.

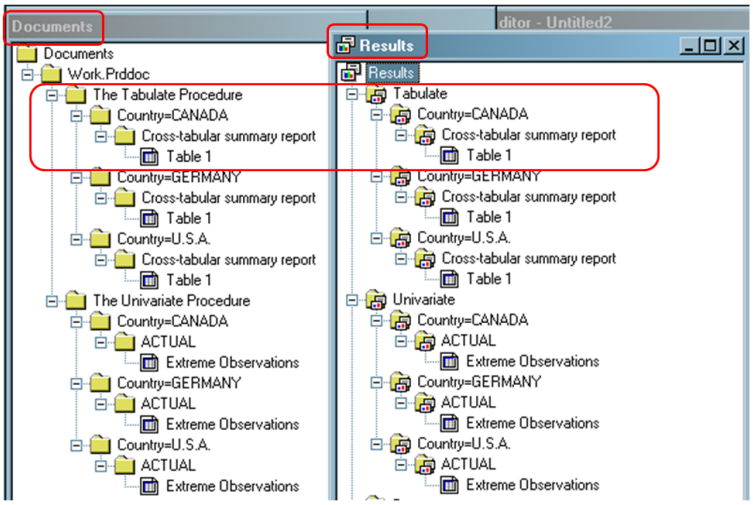


Figure 7

The properties for each item in the document structure are identified as type of "Document", "Folder", or "Table", as shown in this screen shot. In Figure 3, the Folder was references with a type of "Dir", but the reference for "Table" is the same in both Figure 3 and Figure 8. In Figure 3, the name of the ODS WORK.PRDDOC.

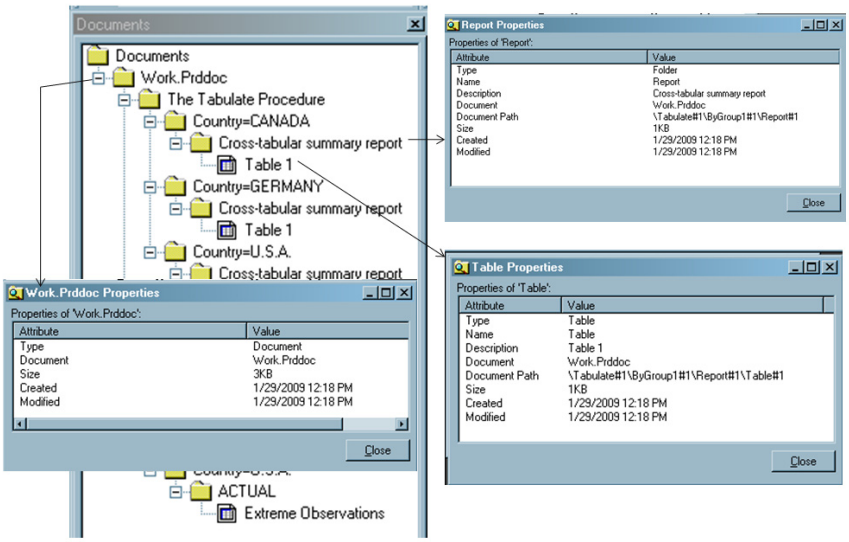


Figure 8

To replay an entire Document store, right-click on the store name and select **Open As…**

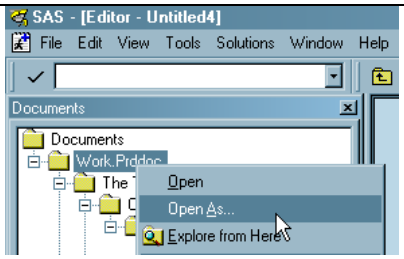


Figure 9

When the Open As window appears, select a destination, such as PDF (highlighted), and then click **OK**.

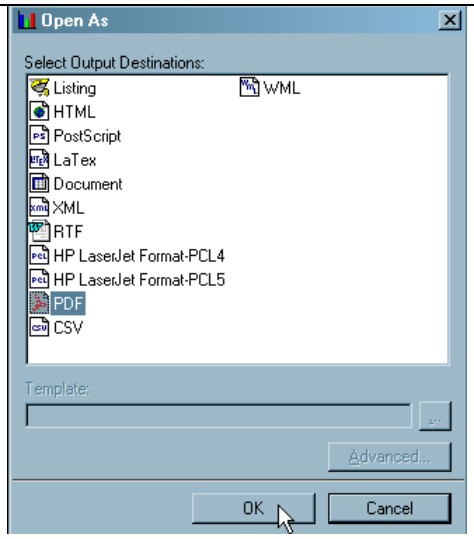


Figure 10

The Open As output is shown in Figure 11. When you use the Open As window, SAS creates a default name, in this instance, SASPDF.PDF. Using PROC DOCUMENT and the REPLAY statement gives you the opportunity to pick the name that you want for your ODS output when you REPLAY the document.

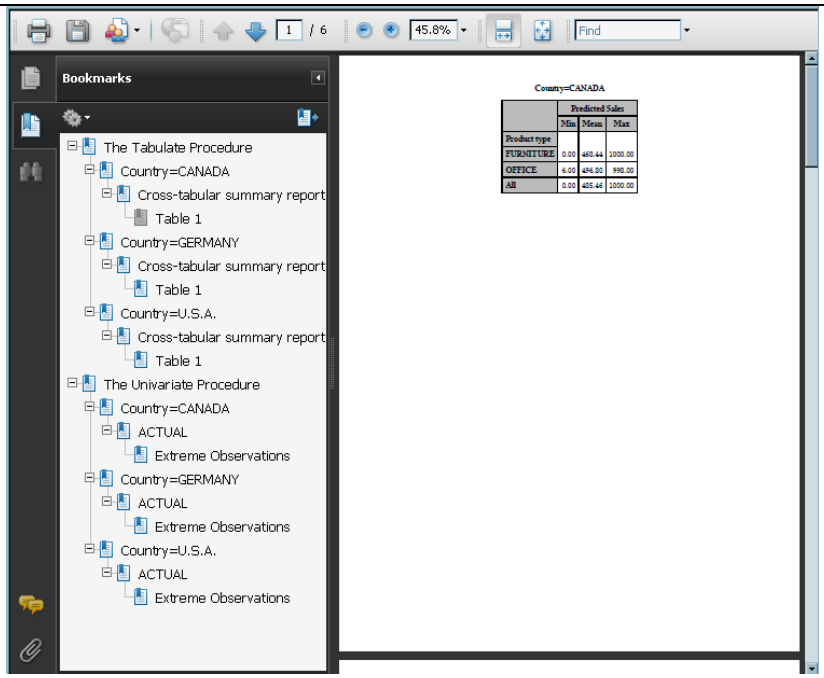
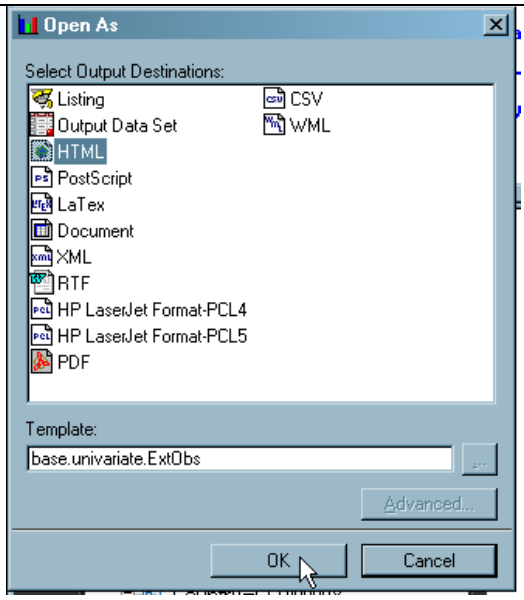


Figure 11

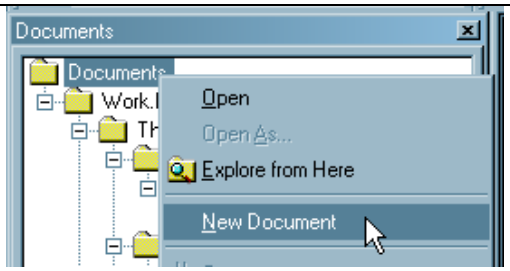
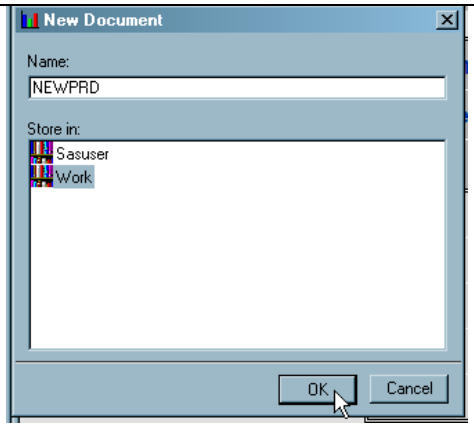
If you highlight only one output object (such as for Extreme Observations) and then select **Open As…,** notice how the name of the Table template for BASE.UNIVARIATE.EXTOBS is specified as the one to use for the object replay.	 **Figure 12**
Since HTML was selected as the destination, when you look in the Results window, the output object that you highlighted (GERMANY, for example) will appear in the Results Viewer.	*The UNIVARIATE Procedure* *Variable: ACTUAL (Actual Sales)* **Country=GERMANY** **Extreme Observations** **Lowest** / **Highest** **Value** **Obs** **Value** **Obs** 3 747 994 652 13 686 996 481 14 710 996 584 15 626 996 863 22 835 1000 663 **Figure 13**

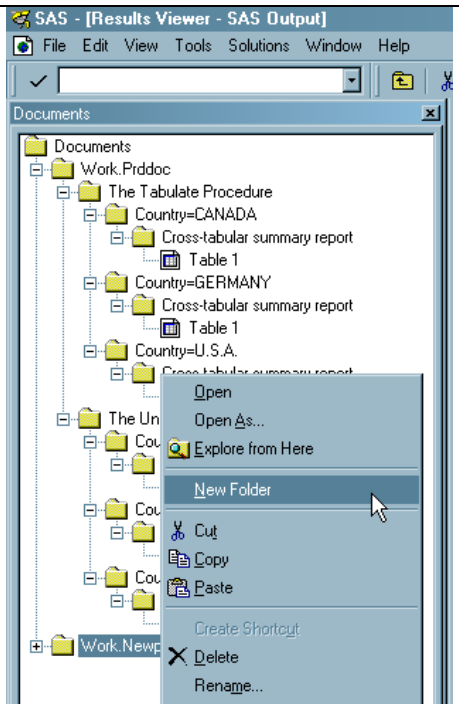
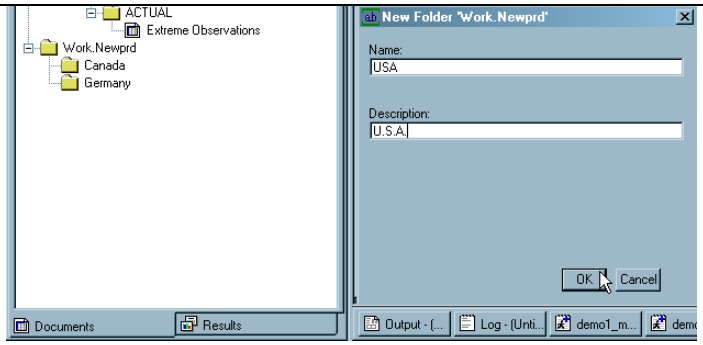
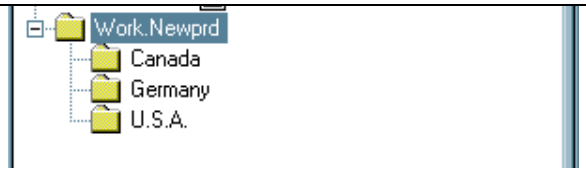
There are other features of the interactive Document window, but all of these features revolve around using the Document window to rearrange the structure of the output objects prior to object replay.

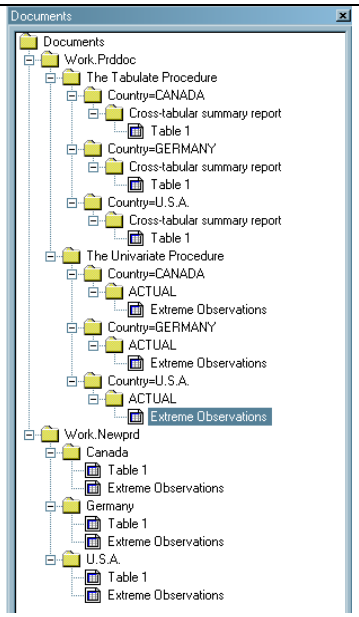
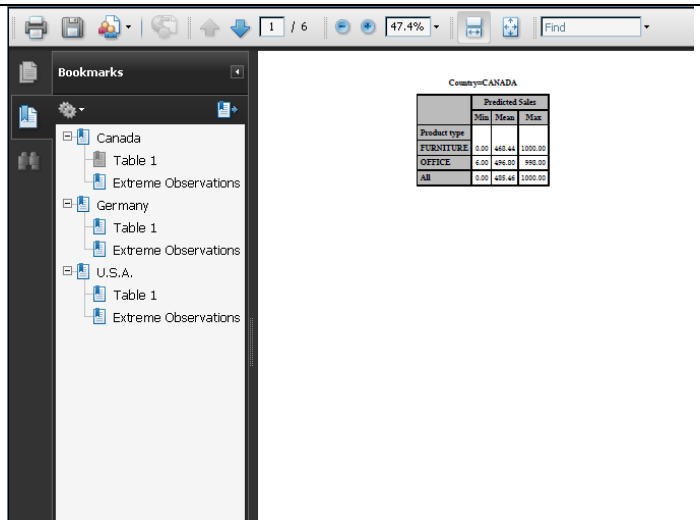
USING THE DOCUMENT WINDOW TO REARRANGE AND REPLAY ODS DOCUMENTS

The next series of figures illustrates creating a new ODS Document in this Document store by using point and click (or drag and drop) actions. Much of what can be done in this interactive mode to rearrange and rename output objects can be done with PROC DOCUMENT syntax.

However, the PROC DOCUMENT syntax actually parallels the process you go through in the interactive window. This means that study of the interactive Document window will enhance your understanding of the PROC DOCUMENT syntax for accomplishing the same end results.

Task and Notes	Figure
To create a new ODS document in this document store, highlight the word **'Documents'** at the top of the Document window and select **New Document** from the menu.	 **Figure 14**
When the New Document window opens, select a library to store the new document store and provide a name for the document store, NEWPRD, for example. Then, click **OK**.	 **Figure 15**
When you start to copy objects from the original document store to the new document store, the document structure is automatically collapsed. It's useful to set the folder to automatically expand. You do this by entering the AUTOEXPAND command into the command area.	 **Figure 16**

To create a folder in the new document store, highlight the name of the new document store, right-click, and select **New Folder**.	 **Figure 17**
Enter the country and the description in the New Folder window and then select OK. Folders for Canada and Germany were already created prior to this screen shot for the creation of the U.S.A. folder. Note that if a Description value is provided, it will be the value used.	 **Figure 18**
The folder with all three countries is shown here. Even though the folders are expanded, they have no objects, so there is nothing to show in the expanded view.	 **Figure 19**

You can either copy and paste or drag and drop from the original document store (WORK.PRDDOC) to the new document store. Figure 20 shows the Output objects after they have been copied to the new document store. You could have moved and rearranged objects in the original document store. However, if your company requires any type of audit trail of output or if you might ever need to reproduce the original report, you should keep an archive copy of the original output structure intact.	 **Figure 20**
Now, if you follow the same steps as those which produced Figures 9, 10, and 11, you could replay the new document store to the PDF destination, for example to get the results as shown in Figure 21. Note how the bookmarks in the PDF output are more streamlined than the bookmarks in the original PDF output, as shown in Figure 11.	 **Figure 21**

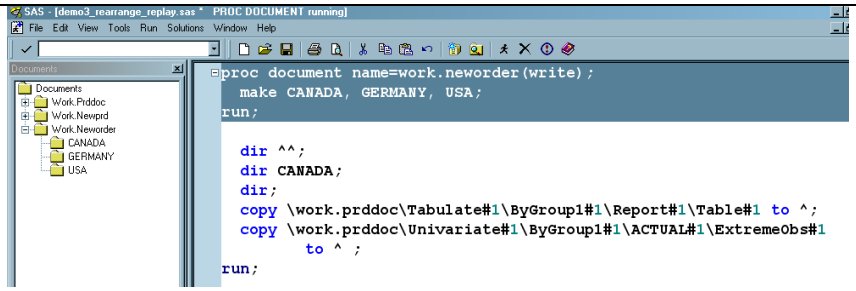
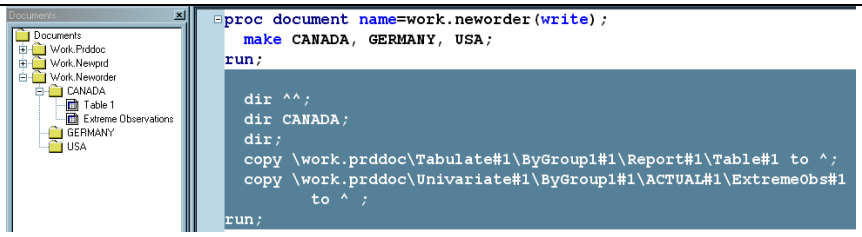
The ODS DOCUMENT window is very straightforward and intuitive to use. We can accomplish exactly the same tasks as those shown in Figure 7 through 21 using PROC DOCUMENT syntax. The next topic will illustrate using PROC DOCUMENT syntax to create, rearrange, and replay an ODS Document store.

Using the GUI window is handy when you have one or two documents you need to create and replay. However, if you work on SAS® Enterprise Guide® (without access to the DOCUMENT window) or you want to create an ODS Document store as part of nightly production processing, then the PROC DOCUMENT syntax will allow you to accomplish these creation, management, and replay tasks.

USING PROC DOCUMENT TO REARRANGE AND REPLAY ODS DOCUMENTS

PROC DOCUMENT is an interactive procedure. As seen in the code to create and list the original ODS document store, you issue PROC DOCUMENT statements, and the RUN statement executes the statements, but, as with other "run time" procedures, PROC DOCUMENT does not use RUN as the step boundary. The step boundary for a PROC DOCUMENT step is the QUIT statement. In order to show how PROC DOCUMENT works, a new document store will be created and then enhanced using the batch PROC DOCUMENT syntax.

One of the coolest things about using the PROC DOCUMENT syntax is that if you have the ODS DOCUMENT window open at the same time as you submit your code, you can watch the new document store as it is populates with objects and folders. This next series of figures illustrates how to use the PROC DOCUMENT syntax to create items in the document store.

Task and Notes	Figure
The highlighted program code creates the new document store, WORK.NEWORDER. The MAKE statement creates three directories or folders, one for each country. The highlighted code was submitted and the results are immediately shown in the open Documents window. Since PROC DOCUMENT supports Run-group processing, you can see the "PROC DOCUMENT running" message at the top of the SAS window.	 **Figure 22**
Next, the DIR statement causes an internal directory navigator to navigate back to the root directory of the document store. The syntax of this statement may look familiar to DOS or UNIX shell programmers who are familiar with navigation using DIR . or DIR .. syntax to navigate up and down the folder paths. Once positioned at the root directory (using ^^ instead of ..), the DIR statement causes navigation to the CANADA directory. If the LISTING destination is open, the current directory location will be echoed in the Output window as a result of the DIR statement before the COPY statement. For	 **Figure 23**

example, the two COPY statements copy objects from WORK.PRDDOC into WORK.NEWORDER. The two objects are shown in Figure 23, after the statements have been submitted.

Similar statements are executed for GERMANY.

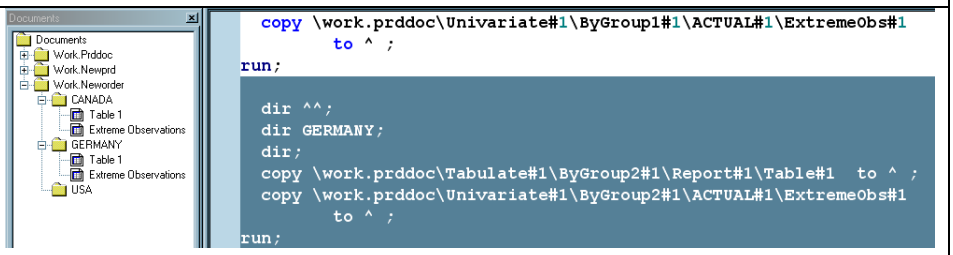


Figure 24

Next, similar statements are executed for USA as shown in Figure 25. The final QUIT statement terminates PROC DOCUMENT execution. The WORK.NEWORDER document store now contains, via program code, the same rearranged objects as WORK.NEWPRD, as seen in Figure 20.

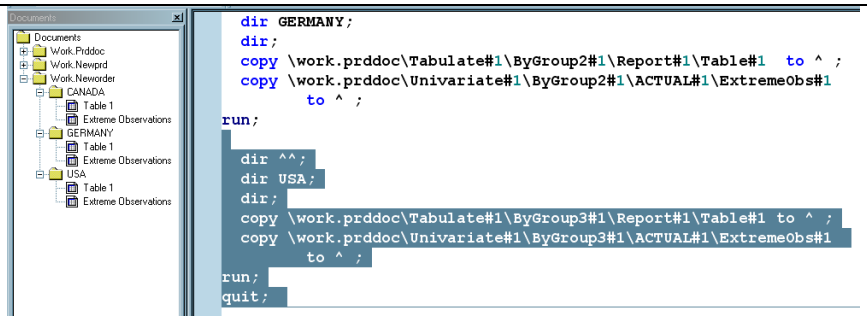


Figure 25

Again, if you highlight the document store name and select Open As…, then you can replay the new document store to open destinations, as already illustrated. However, you can also submit PROC DOCUMENT code to REPLAY an ODS document, as shown next to Figure 26. Since the LISTING destination was also open, the REPLAY statement also replayed the document store to the LISTING, HTML, and PDF destinations.

```
ods html file='replay_new.html'
         style=sasweb;
ods pdf file='replay_new.pdf';
proc document name=work.neworder;
   replay;
run;
quit;

ods _all_ close;
```

REPLAY Code

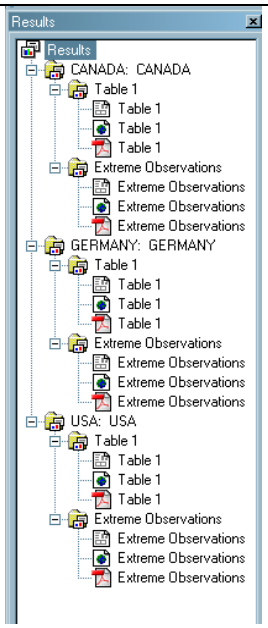


Figure 26

The PDF output is shown in Figure 27 and shows the same structure in the bookmark area as that created in Figure 21 with the Open As method.

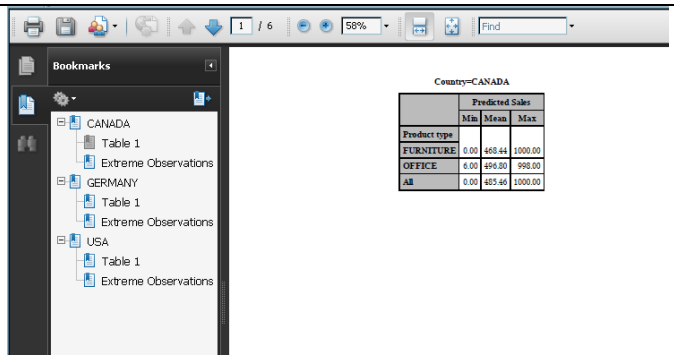


Figure 27

In addition to replaying all of a document store, you can use other features of the REPLAY statement to replay using a directory reference or using WHERE expressions. For example, to replay only the PROC TABULATE output, recall that the name of the output objects for PROC TABULATE in our original document store all contained the string "Table#1" as shown in Figure 3. If you submit a PROC DOCUMENT step with the LIST statement to see the entries in WORK.NEWORDER, this same object name (Table#1) is still in use, even though the directory name reflects the new structure, as seen in Figure 28.

Number of levels: All

Obs	Path	Type
1	\CANADA#1	Dir
2	\CANADA#1\Table#1	Table
3	\CANADA#1\ExtremeObs#1	Table
4	\GERMANY#1	Dir
5	\GERMANY#1\Table#1	Table
6	\GERMANY#1\ExtremeObs#1	Table
7	\USA#1	Dir
8	\USA#1\Table#1	Table
9	\USA#1\ExtremeObs#1	Table

Figure 28

The following REPLAY statements can replay entries using WHERE processing or directory reference.

```
** Replay Using a WHERE expression;
ods pdf file='c:\temp\replay_where.pdf';
proc document name=work.neworder;
   replay ^(where=(_path_ contains 'Table#1')) ;
run;
quit;
ods _all_ close;

** Replay a directory by directory name;
ods pdf file='c:\temp\replay_dir.pdf';
proc document name=work.neworder;
   replay \CANADA#1 / levels=all;
run;
quit;
ods _all_ close;
```

These document replays were sent to the PDF destination so that the PDF bookmark area could be scanned to show the how the selection criteria was used to make the ODS output file. The replays could have been sent to other destinations as well. The output from the first PROC DOCUMENT step is shown in Figure 29 and from the second PROC DOCUMENT step in Figure 30. This ability to select output objects for replay, allows you to pick and choose entries from the document store without rerunning the original procedure code. This is particularly useful for those ad hoc requests that start out like this: "You know that report you do that has all three countries? Is there any way to just get the TABULATE stuff for each country." Or: "You know that report you just sent? Could I only get the info for CANADA?"

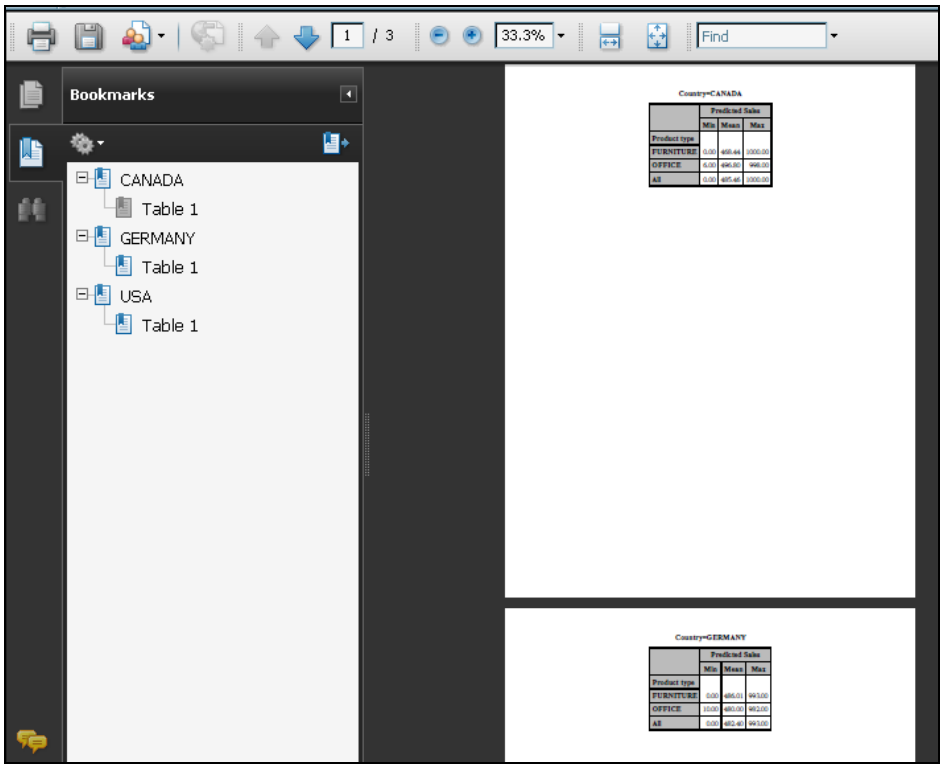


Figure 29: Partial ODS PDF Results Shown in Acrobat Reader

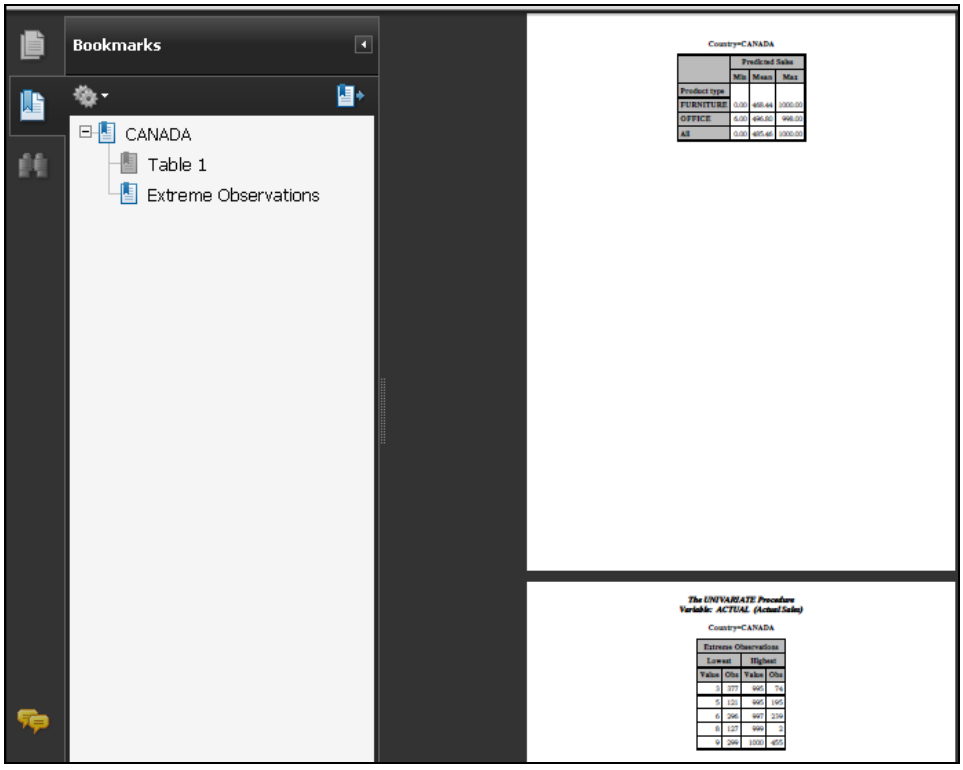


Figure 30: Partial ODS PDF Results Shown in Acrobat Reader

The WHERE expression processing is possible starting in SAS® 9.2 because ODS DOCUMENT now surfaces a special group of subsetting variables that you can use in WHERE expressions with the following PROC DOCUMENT statements: COPY TO, MOVE TO, LIST, REPLAY, and DELETE. Some of these variables are shown in Table 1. Refer to the documentation for the complete list.

Subsetting Variable	Meaning
CDATE	creation date of the current entry
LABEL	label of the current entry
LABELPATH	path to the label of the current entry
MDATE	modification date of the current entry
NAME	name of the current entry
PATH	path of the current entry
TYPE	type of the current entry (Table or Graph)
Variable-name	name of a BY variable

Table 1: Partial List of Subsetting Variables for WHERE Expressions

The previous examples showed the use of WHERE processing and directory specification for the REPLAY. Recall that our original document was created with By Group processing. This means that COUNTRY is a valid subsetting variable for a WHERE expression, for REPLAY purposes. This REPLAY code for GERMANY represents an alternative to the directory method used for Figure 30.

```
** Replay Using a BY variable name;
ods pdf file='c:\temp\replay_var.pdf';
proc document name=work.neworder;
   replay ^(where=(Country = 'GERMANY')) ;
run;
quit;

ods _all_ close;
```

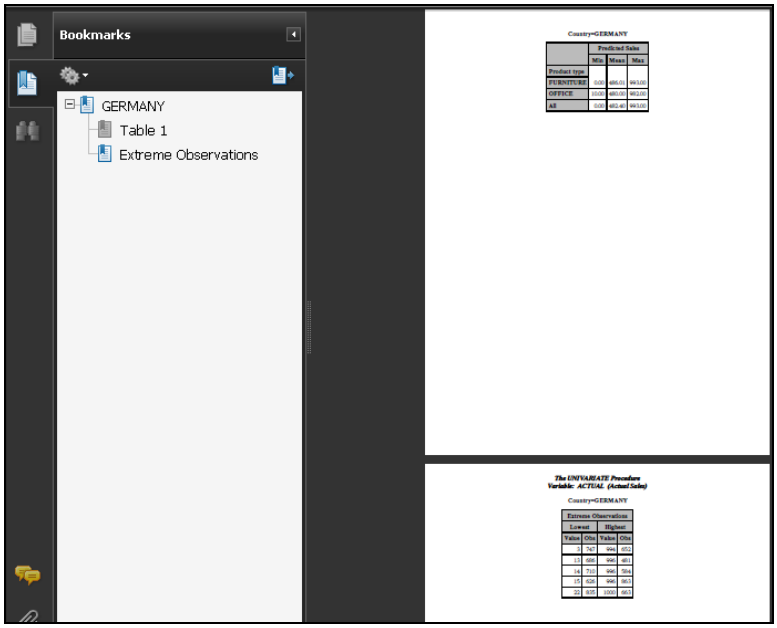


Figure 31: Partial PDF Output Viewed in Acrobat Reader

Do not be confused by the ^ symbol in WHERE expression. Remember that ^ is the indicator for the current path location in the document. The indicator for the parent path is ^^. This convention is almost exactly like the DOS navigation commands. Since the REPLAY statement is capable of either replaying the entire document or part of a

document, it's necessary to precede the WHERE expression for REPLAY with a path indicator. If you are replaying an entire document, the REPLAY statement does not need a path indicator. However, if you are going to use a WHERE expression, then a REPLAY statement must have a path indicator, even if it is just the ^ to indicate every object under the current path is subject to the WHERE expression selection criteria.

A more advanced use of the REPLAY statement is the ability to direct output from different REPLAY statements to different destinations.

```
** Replay to Multiple Destinations;
ods html file='c:\temp\replay_dest.html'
         style=sasweb;
ods pdf file='c:\temp\replay_dest.pdf';
ods rtf file='c:\temp\replay_dest.rtf'
        style=journal startpage=no;
proc document name=work.neworder;
   replay \USA#1 / levels=all dest=(html pdf);
run;

   replay ^(where=(_path_ contains 'Table#1'))/ dest=rtf ;
run;
quit;
ods _all_ close;
```

As shown by the output in Figure 32, different REPLAY statements can use the DEST= option to route different output objects to different destinations. The RTF output shown on the left in Figure 32 only shows the three PROC TABULATE outputs and the HTML output on the right in Figure 32 shows the TABULATE output followed by the UNIVARIATE output for U.S.A.

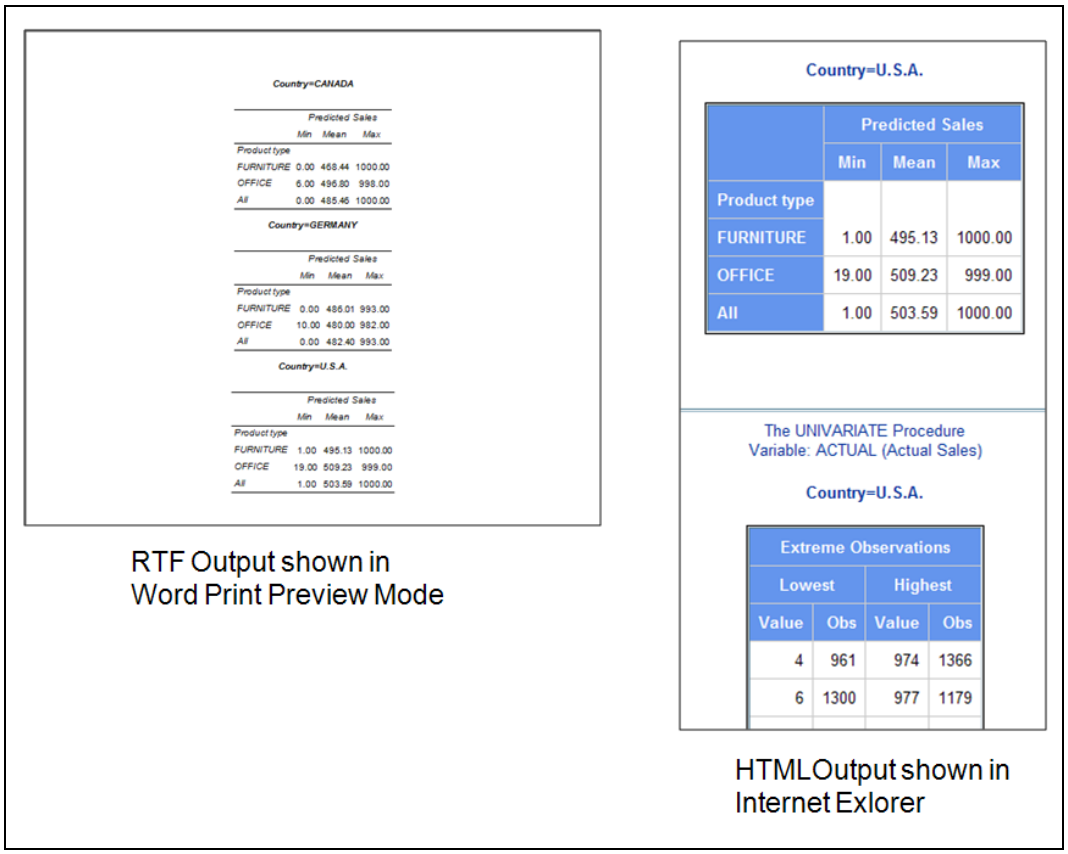


Figure 32: Different REPLAY Statements Sent Output Objects to Different Destinations

Note how the STARTPAGE=NO option was specified for the ODS RTF output file, which resulted in the page breaks being suppressed between the PROC TABULATE tables. However, since the BYLINE information was part of the

saved information for each output object, the bylines appear above each table. The PDF output is the same as the HTML output, as shown in Figure 33. Since STARTPAGE=NO was not specified for the PDF destination, you can see that the UNIVARIATE output for U.S.A. appears on a separate page.

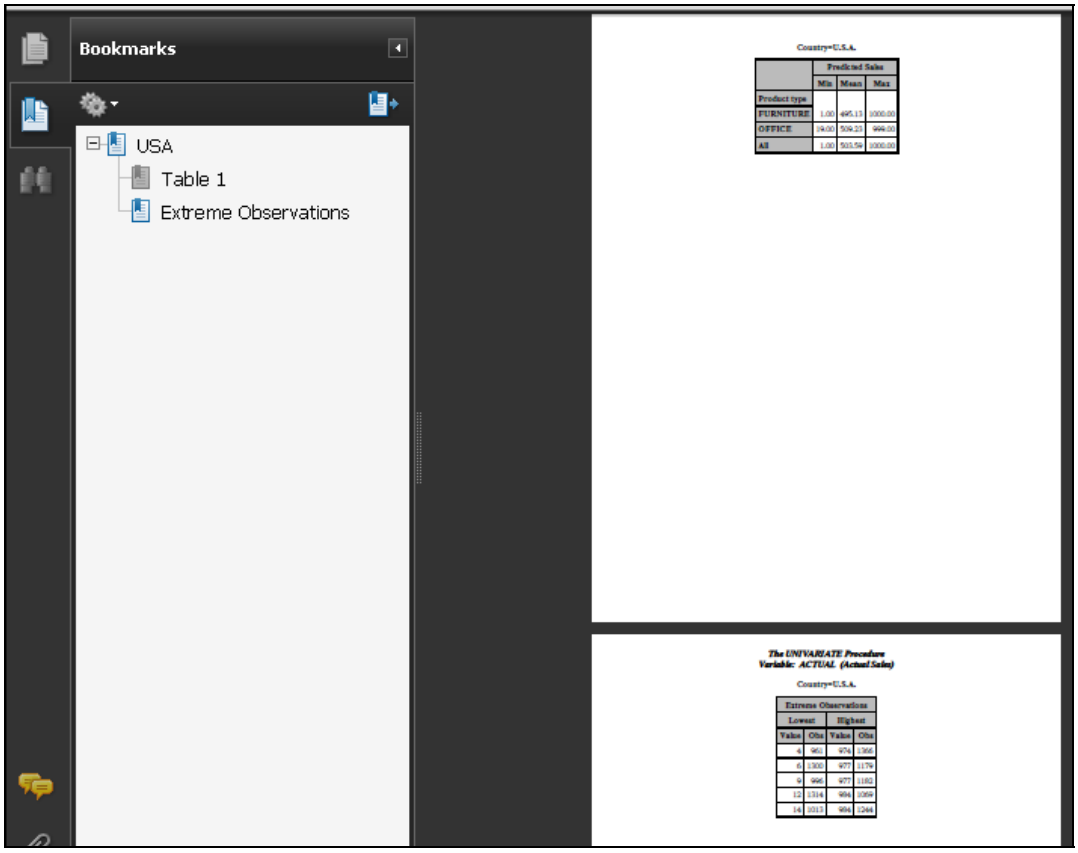


Figure 33: PDF Output from REPLAY with DEST= Option

FURTHER ENHANCEMENTS USING PROC DOCUMENT

The ability to use WHERE expressions and directory processing allows you to alter which output objects you get in the replayed destination output, but so far, the original output objects have only been rearranged and replayed. The output objects are still the same as they were in the original WORK.PRDDOC. There are, however, some further enhancements that you might want to make. For example, the current report (in the PDF destination) is six pages long. There is enough room on the TABULATE output page to put the Extreme Observations output on the same page. This would cut the report from six pages to three pages – one page for each country. Or, you might want to insert some text specific to each country. Enhancements like these can only be accomplished with PROC DOCUMENT, although the ability to perform these tasks is envisioned in upcoming releases of ODS DOCUMENT.

To show you how to make these types of enhancements to an ODS Document, the next topic will use PROC DOCUMENT syntax to create, manage, and enhance the ODS Document store. There are some actions, as shown in Table 2, that you can use either the Document window method or PROC DOCUMENT to accomplish. However, as seen in Table 3, there are some actions that can only be accomplished with the PROC DOCUMENT syntax method.

We have only worked with the LIST, DIR, MAKE, COPY, and REPLAY statements in previous code examples. When working with an existing document store, it will be necessary to work, again, with the object names as shown by the LIST statement in Figure 28.

The enhancement statements are very straightforward. The OBPAGE statement allows page breaks to be inserted or deleted. The OBBNOTE allows the insertion of notes before an output object, while the OBANOTE allows the insertion of notes after an output object (this example code uses only 'before' notes and titles). The OBBNOTE statement for our output objects will appear under the first byline on the page. When the page break for the Extreme Observations object is deleted, it will also delete the byline information for the Extreme Obs object.

Tasks That You Can Perform with Both Methods	Documents Window	Document Procedure
Create a new ODS document.	Yes	Yes
Create a new folder.	Yes	Yes
Copy folders or output objects.	Yes	Yes
Move folders or output objects.	Yes	Yes
Create a symbolic link from one output object to another output object.	Yes	Yes
Delete a document, folder, or output object.	Yes	Yes
Rename a folder or output object.	Yes	Yes
Assign a description to a folder or output object.	Yes	Yes
Prevent entries from being displayed when they are replayed.	Yes	Yes
Show entries that are excluded.	Yes	Yes
Enable hidden entries to be displayed.	Yes	Yes
Replay to the specified open ODS destinations.	Yes	Yes
Determine the path specification.	Yes	Yes

Table 2: Tasks Accomplished with ODS DOCUMENT Window and PROC DOCUMENT

Tasks That You Can Only Perform with PROC DOCUMENT	Documents Window	Document Procedure
Import a data set or graph segment.	No	Yes
Set or display the current directory.	No	Yes
Create or delete a page break.	No	Yes
Create or modify title lines.	No	Yes
Create or modify subtitles.	No	Yes
Create of modify the lines of text before output objects.	No	Yes
Create or modify the lines of text after output objects.	No	Yes
Create or modify footnote lines.	No	Yes
Create text strings in the current folder.	No	Yes

Table 3: Tasks That Are Only Accomplished with PROC DOCUMENT

The code to make changes to WORK.NEWORDER is shown below. The DIR command at the top of the program is used to ensure that the path navigator is positioned at the top of the document store.

```
proc document name=work.neworder(update);
  dir ^^;
  obpage \CANADA#1\ExtremeObs#1 /delete;
  obbnote \CANADA#1\Table#1 'Something' /just=center;
  obbnote \CANADA#1\ExtremeObs#1 'Text1' / just=center;
  obpage \GERMANY#1\ExtremeObs#1 / delete;
  obbnote \GERMANY#1\Table#1 'Something Else' /just=center;
  obbnote \GERMANY#1\ExtremeObs#1 'Text2' / just=center;
  obpage \USA#1\ExtremeObs#1 /delete;
  obbnote \USA#1\Table#1 'Another Thing' /just=center;
  obbnote \USA#1\ExtremeObs#1 'Text3' / just=center;
  setlabel \CANADA#1\Table#1 'Min, Mean, Max';
  setlabel \GERMANY#1\Table#1 'Min, Mean, Max';
  setlabel \USA#1\Table#1 'Min, Mean, Max';
run;
quit;
```

Figure 34 shows the Document window view of WORK.NEWORDER after the PROC DOCUMENT statements were submitted. In addition to the Document window, the Properties window for the first Table for Canada is shown.

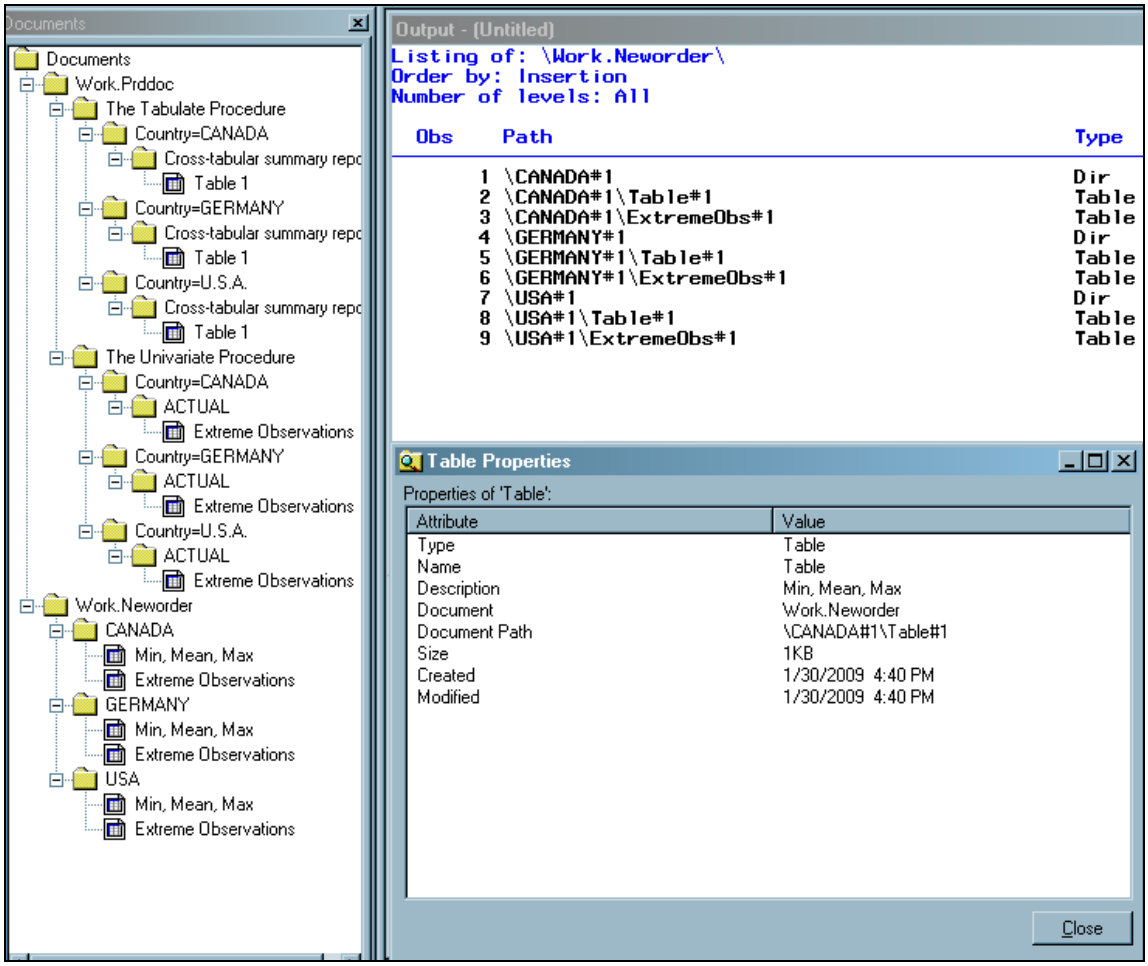


Figure 34: WORK.NEWORDER Displays New Labels

The SETLABEL statement changes the description that you see in the Document window. However, if you ran a LIST statement, the names of the output objects would be the same as those seen in Figure 28. Now, when you REPLAY the WORK.NEWORDER document to the PDF destination, note how it is the object labels that appear in the PDF bookmark list as shown in Figure 35.

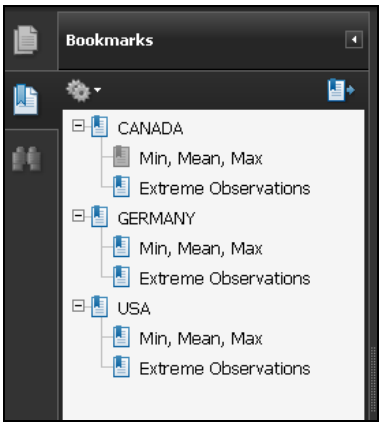


Figure 35: ODS PDF Bookmarks Use New Labels

If you look at the page for Canada, shown in Figure 36, next to the page for Germany, you can see the impact of the OBBNOTE statements.

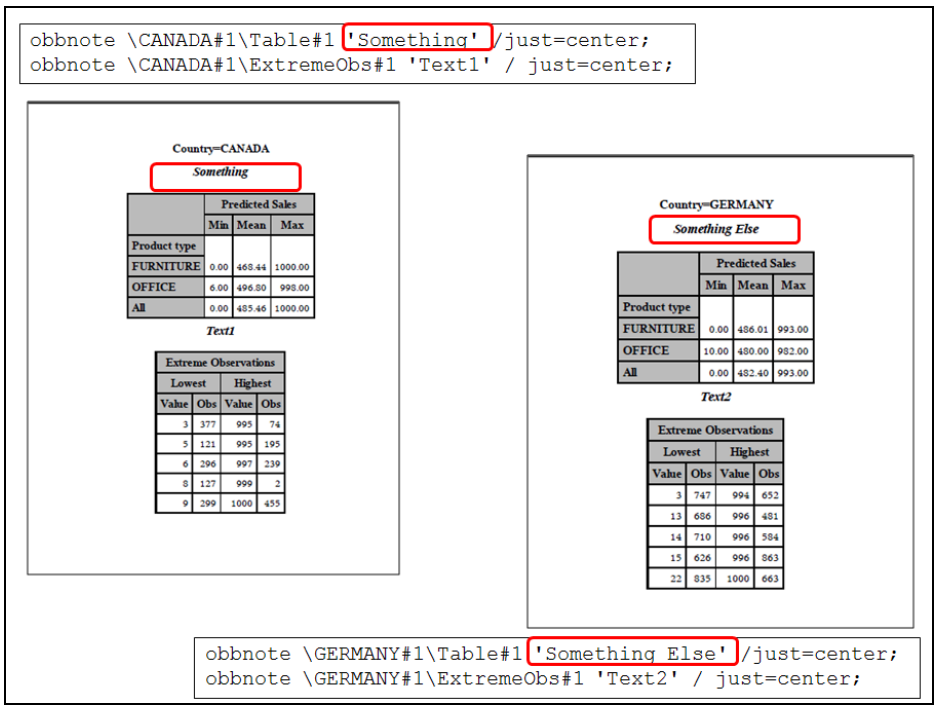


Figure 36: Results of OBBNOTE Statement in the PDF Results

The OBTITLE statement (although not used here) allows different titles to be inserted for each output object, in the same way that we used OBBNOTE to insert a note before each output object. This makes it possible for a page break associated with an output object to have a different title from a page break for a different output object. Or, in more common usage, Canada's page break or output objects could have a different title from Germany's objects. If you sent the output objects from the document store to the HTML destination, the horizontal rule between output objects (the page break indicator) is gone, indicating that the OBPAGE statement was able to have an impact on the HTML destination, as well as the PDF destination, as shown in Figure 37.

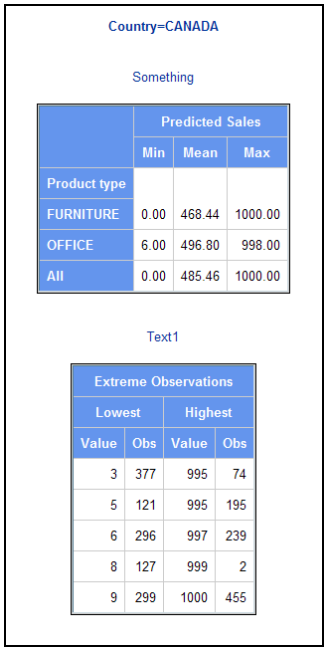


Figure 37: Results of OBBNOTE Statement in the HTML Results

You can store more than tables in an ODS DOCUMENT store. The primary entries kept in the document store are table objects, graph objects, and notes. The equation object, mentioned in the documentation, is used internally by some procedures but is not an entry type that can be managed or manipulated under user control. There are more details to learn about using ODS DOCUMENT stores and the entries that can be stored in them; however, this paper cannot cover them all. The easiest way to get started with ODS Document stores is to create some output and experiment with interactively rearranging your objects in the Document Window. From that experience, you can venture out into PROC DOCUMENT syntax.

CONCLUSION

This paper has only touched the surface of what you can do with ODS documents. Further study of ODS DOCUMENT and PROC DOCUMENT syntax will allow you to present your reports in a form and structure dictated by the report needs, and not by the order of the procedures that created the output objects. ODS DOCUMENT allows you to exercise an unprecedented level of control over your reports, allowing you to have your reports, your way, with the structure and object names of your choice. In addition, the ability to freeze or store output objects as they were originally created by SAS and ODS gives you an archive and replay framework that is an improvement over past methods, such as storing copies of the LISTING reports or copies of the HTML, RTF, or PDF files.

ACKNOWLEDGMENTS

The author thanks David Kelley for graciously answering all ODS DOCUMENT and PROC DOCUMENT questions. In addition, many thanks are owed to Amy Wolfe, who was instrumental in editing this paper to ensure that it was free of grammatical errors, in spite of the author's best attempts to the contrary.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Cynthia L. Zender
SAS Institute, Inc.
Work Phone: 575-522-3803
E-mail: Cynthia.Zender@sas.com