

PROC DOCUMENT by Example Using SAS®

Michael Tuchman

SAS® Press



Contents

About This Book	xiii
About The Author	xvii
Acknowledgments	xix
Chapter 1: Introduction and Quick-Start Guide	1
What Is an ODS Document?.....	1
Life Before ODS Document	2
Life with ODS Document	2
Why Use an ODS Document?	2
Why Not Manage Change with Office Software?.....	3
Quick-Start Guide.....	3
File Location.....	3
Data.....	4
Saving Output to an ODS Document.....	4
Replaying Output	5
Subsetting Output	8
Managing Output.....	10
Changing Titles and Footnotes.....	12
Using Folders	13
Deleting and Overwriting Document Contents	15
What Remains to Be Covered?.....	16
Summary	16
Chapter 2: The ODS DOCUMENT Destination	17
Introduction	17
Convention.....	18
What Is the Output Delivery System (ODS)?.....	18
Definition of the ODS Destination.....	18
ODS without the Document Facility	19
ODS with the DOCUMENT Destination.....	21

Role of the DOCUMENT Procedure.....	23
Reasons to Use the Document.....	24
Naming ODS Output.....	24
The ODS TRACE Statement	25
Determining the Directory Where ODS Output Is Stored	27
Implementation Details of the Item Store.....	28
Controlling Output Using SELECT and EXCLUDE Lists.....	28
Summary	30
Chapter 3: The ODS DOCUMENT Statement.....	31
Introduction	31
The ODS DOCUMENT Statement	32
The NAME= Option.....	32
Example: Storing Output at the Top Level.....	32
Comparing ODS TRACE Output to ODS Document Paths	36
The DIR= Option	36
Behavior of the Pathname	38
The CATALOG= Option.....	40
Using Access Options (Write and Update)	40
Example: Reviewing Your Progress	42
Summary	44
Chapter 4: Listing Documents Using the DOCUMENT Procedure.....	45
Introduction	46
The DOCUMENT Procedure.....	46
Terminology	46
Directory Notation	47
Root Level Directory	47
Current Directory.....	47
Invoking the DOCUMENT Procedure	47
The NAME= Option.....	48
The LABEL= Option.....	50
Running the Document Procedure Interactively.....	50
Access Modes	51
The LIST Statement.....	51
Simple Invocation of the LIST Statement	52
Sequence Numbers	53

LEVELS= Option	56
Using the DETAILS= Option	60
ORDER= Option	62
BYGROUPS Option.....	63
Warning: BYGROUPS Option Might Show No Output.....	66
Using WHERE= Clauses to Conditionally View a Document.....	67
WHERE Variables	67
Examples.....	68
Interlude	70
Debugging WHERE= Clauses.....	71
The WHERE= Clause and ODS Names	73
Comparison with the ODS SELECT and EXCLUDE Statements	73
LABELPATH - Proceed at Your Own Risk.....	74
The SETLABEL Statement.....	75
Important: Labels Are Not Titles.....	77
Summary	79
Chapter 5: The REPLAY Statement	81
Introduction	81
How Replay Works.....	82
REPLAY Statement Syntax.....	83
Replaying to All Open Destinations.....	84
The DEST= Option.....	86
Replaying to Multiple Destinations.....	87
Replaying in a Different Order.....	88
The LEVELS= Option.....	88
Overriding Titles with the ACTIVETITLE and ACTIVEFOOTN Options.....	89
Replaying Using WHERE= Clauses	89
Replaying Subsets of Output Objects	90
Example: Reordering BY-Grouped Output	95
Replaying Output with Temporary Formats	96
Temporary or Permanent Formats?	98
Summary	100
Chapter 6: Managing Folders	103
Introduction	104
How ODS Document Folders are Different	104
Absolute Pathnames	104
Relative Pathnames	105

Positioning Arguments	105
When the TO Argument Requires a Sequence Number	106
WHERE= Clauses and Directories	106
The MAKE Statement.....	107
The DIR Statement	110
The RENAME TO Statement	112
Renaming a Document	112
Warning about Renaming Output Objects.....	114
The COPY TO Statement	115
A Common Mistake	117
Copying Using the LEVELS= option	123
The MOVE TO Statement.....	124
Browsing Collections of Documents with the DOC Statement.....	126
The LIBRARY= Option.....	127
The NAME= Option.....	128
The DELETE Statement	129
Warning: You Cannot Delete the Current Directory	130
The HIDE Statement.....	132
Extended Examples	136
Replay with the LEVELS= Option.....	136
Listing Every Document in Your System.....	140
Description of %alldocs Macro Arguments	141
Managing ODS Documents via the Graphical User Interface.	143
Summary	143
Chapter 7: Customizing Output.....	145
Introduction	145
The Document Used in This Chapter	146
Output Objects in an ODS Document	146
Introduction to Context	148
Example: A Context Reference Chart	148
The OBPAGE Statement	151
The OBPAGE Statement Might Be Ignored in Some Procedures.	155
The OBTITLE Statement	156
The SHOW option.	159

The OBSTITLE Statement.....	161
Using #BYVAL and #BYVAR Variable Values to Build a Custom Subtitle	163
Example: Using the #BYLINE Variable Value.....	164
Object Notes: The OBANOTE and OBBNOTE Statements	165
Document Notes	167
Adding Notes Using the TEXT Statement.....	167
Adding Notes Using the DOCUMENT Procedure’s NOTE Statement.....	170
Document Notes Compared with Object Notes	173
The Final Version of the Example Document	174
Summary	176
Chapter 8: Exporting to Data Sets.....	177
Introduction	177
The Power of Combining the ODS OUTPUT Destination with the Document Facility	178
The ODS OUTPUT Destination.....	179
The ODS OUTPUT Statement	179
The ODS OUTPUT Destination without the Document	179
The ODS OUTPUT Destination with the Document.....	183
Combining Data Sets from the Document.....	186
Combining the OUTPUT and DOCUMENT Destination.....	191
ODS OUTPUT and the DOCUMENT Procedure’s Data Sets.....	192
Simple List Output.....	192
Contents of the Document Data Listing	192
Conditionally Executing the DOCUMENT Procedure with CALL EXECUTE	194
The SASDOC Engine.....	197
Accessing a Directory of Output.....	198
Applications of the SASDOC Engine.....	200
The SASDOC Engine and Sequence Numbers.	205
Summary	210
Chapter 9: Importing Data to the Document	211
Overview.....	211
The Document Is Not a SAS Library.....	212
The IMPORT Statement.....	212
Data for Examples	212
Importing a Data Set.....	213

Importing Text Files	215
Importing Data and Its Template to a Document	218
Summary	218
Chapter 10: Working with Links	219
Introduction	219
Creating a Link	220
Source and Target.....	221
Replaying Links	221
The FOLLOW Option of the LIST Statement.....	222
Orphaned Links	223
Document Management Using Links.....	224
Modifying PDF Bookmarks Using Links	227
Hard Links	228
Summary	233
Chapter 11: Working with Templates	235
Introduction	235
What Is a Template?	236
Terminology	237
Template Store	237
Template Search Path.....	237
Template Access Mode	237
Updated Document for Examples in This Chapter	238
Defining Your Own Template Store.....	238
Template Store Organization	240
Exploring the Template System.....	240
Example: Removing Automatic Page Breaks	244
Modifying Template Definitions	247
Undoing Template Changes.....	248
Templates and the Document.....	249
Identifying Templates in the Document Listing	249
The OBTEMPL Statement.....	250
Document Saves a Reference, Not the Actual Template.....	251
Modifying the DOCUMENT Procedure's Templates.....	256
Styles	258

Assigning a Custom Template to an Object in a Document.....	258
The Custom Template.....	259
Data Set Used	259
Summary	262
Appendix	263
References	277
Index	279

Chapter 1: Introduction and Quick-Start Guide

What Is an ODS Document?	1
Life Before ODS Document	2
Life with ODS Document	2
Why Use an ODS Document?	2
Why Not Manage Change with Office Software?.....	3
Quick-Start Guide	3
File Location	3
Data	4
Saving Output to an ODS Document	4
Replaying Output.	5
Subsetting Output.....	8
Managing Output	10
Changing Titles and Footnotes	12
Using Folders	13
Deleting and Overwriting Document Contents	15
What Remains to Be Covered?	16
Summary	16

What Is an ODS Document?

ODS stands for Output Delivery System, which is a system for producing SAS output in a variety of formats such as HTML, PDF, XML, LaTeX, as well as many others.

An ODS document, together with the DOCUMENT procedure, comprises a facility that enables you to store ODS output and reprint it at a later time. The goal of this chapter is to show you why this is such a powerful ability.

To see how easy it is to get started using the DOCUMENT procedure, review the examples in the quick-start guide in this chapter.

2 PROC DOCUMENT by Example Using SAS

This ability to reproduce output without re-running SAS code is at the heart of the Document Facility. To see why this is useful, consider the following two scenarios.

Life Before ODS Document

Suppose your boss wants a summary of last week's results for the company's intranet site, and wants them displayed using the company's style standards. To reproduce this output, you try to find the analysis data set that was used to create the original results. You might need to search among many similarly named data sets in order to find the right one. Then, you need to find the version of the code that created the published results. You hope that this is the latest version of your code, but it might not be. Although you can re-run the process, you feel concerned about spending valuable time duplicating existing work.

Your boss arrives at your cubicle and asks whether you are ready. "Just a few more minutes," you timidly reply, wishing that you had been ready earlier.

Life with ODS Document

Now, imagine this scenario: When you ran your SAS program originally, you had the foresight to save all of your procedure and graph output to an ODS document. You did this knowing that you might need to access the same output in the future without re-running your SAS program. After all, changing a format in a table or making some changes to a graph's appearance doesn't change any of the underlying data; it is purely a cosmetic change.

Returning to the storyline, now when your boss asks you for your previous results in a different format, you can head directly to your ODS document. Using the ODS HTML destination with the CSSSTYLE= option set to the latest company style sheet, you invoke the DOCUMENT procedure. You re-create all of your original results no matter whether they were produced two weeks ago or two years ago. Furthermore, the output objects in an ODS document can be labeled and dated so you know you have the correct output without the guesswork.

Your boss arrives at your cubicle and sees the desired results already on your screen. There's a smile on your face, a smile on your boss's face, and everyone lives happily ever after.

Why Use an ODS Document?

As a project grows, the ODS Document Facility provides you with a method to conveniently store SAS procedure output, graphs, and data sets all in one location, the ODS document.

As suggested in the second scenario, here are some of the things that you can do with the ODS Document Facility:

- Replay output in an order different from that with which it was originally stored
- Search through and filter an ODS document by date, description, or output type
- Instantly turn these results into professionally styled ODS output
- Add notes about your output
- Change titles, footnotes, and create additional annotative comments

And of course, all of this is done—with the soon-to-be-familiar refrain—without re-running the original code that produced the output.

The ODS Document Facility is an excellent tool when you want to provide output to external clients, vendors, third parties, etc., who must be assured that your results are stored in the document as they were when first produced. This is particularly important for auditing or regulatory submission in the financial and pharmaceutical industries.

Why Not Manage Change with Office Software?

You can also manage your output by cutting and pasting SAS output to commercial office software, such as Microsoft Office or Open Office. If you choose to manipulate output using the third-party software to modify the appearance of your results, you may create versioning problems between your delivered work and your SAS output. If you use the ODS Document Facility, however, you can date and label your output, keep track of multiple versions, and search for output meeting particular criteria, all within the familiar SAS environment.

Quick-Start Guide

By now, you should be convinced of the value of the DOCUMENT procedure and be ready to start using it. The following series of examples shows you how to get started with the ODS Document Facility. You will learn how to create a document, replay the document, as well as create a table of contents and customize the order in which the output appears in a document.

Attempting to cover a lot of ground with a few well-chosen examples does impose some limitations. In order to keep the discussions short, some exceptions to the rules and some subtleties are deferred to the appropriate chapter.

File Location

Since individual file systems differ, filenames in this quick-start guide do not include a directory name in the file path. To make the files easier to find, save the files to a directory of your own choosing, and include this directory in your file path. You will see this written in the quick-start examples:

4 PROC DOCUMENT by Example Using SAS

```
...file "cars.html"
```

When you see that, write instead:

```
file "your-favorite-path-for-saving-files\cars.html";
```

Also, you should define a permanent library named MYLIB in which to store your ODS documents and make sure this definition persists between sessions.

Data

The data set that is used is SASHELP.CARS, excluding the three observations that contain TYPE='Hybrid.'

Program 1.1: Setting Up the Data

```
libname mylib "mylib-library-directory-name";
proc sort data=sashelp.cars out=WORK.cars;
  /* nothing against Hybrids, I promise! */
  where upcase(type) ne 'HYBRID';
  by origin type;
run;
```

Saving Output to an ODS Document

Saving documents is as simple as putting the ODS DOCUMENT NAME= statement before your code and the ODS DOCUMENT CLOSE statement at the end. The argument that follows the NAME= is the name of the document in which you plan to save your output. You must always provide this.

Program 1.2 assumes that there is not already a document named mylib.quickstart. If there is, change the definition of the MYLIB library so that you do not overwrite or mess up any existing work.

The ODS DOCUMENT statement in Program 1.2 is an example of opening an *ODS destination*. An ODS destination typically receives output and produces a version of that output in a particular file format. This format is then displayed with another application, such as a web browser, PDF reader, word processor, or other program capable of reading the output file. When you send output to an ODS document, however, the document is not displayed, but saved in a special file called a SAS *item store*. Instead of reading the output file with a third-party application, you read it with the DOCUMENT procedure.

Program 1.2: Saving Output to an ODS Document

```
ods document name=mylib.quickstart;
proc contents data=cars;
run;
proc univariate data=cars;
by origin type;
run;
proc means nmiss mean max min data=cars;
```

```

class origin type;
run;
ods document close;

```

If you want to send additional output to the document, you run another block of code using the same technique as shown in Program 1.3. The new output will be placed at the end, after existing output.

Program 1.3: Adding Additional Output to an ODS Document

```

ods document name=mylib.quickstart;
proc freq data=sasuser.cars;
  table origin * type / list;
run;
ods document close;

```

Note that this program adds the output from the FREQ procedure to the end of the document. What if you wanted to add it to the beginning of the document? You can move folders to any position within a document with the MOVE TO statement, which will be discussed in Chapter 6, “Managing Folders.”

At this point, you should have a document with output from the CONTENTS, UNIVARIATE, and FREQ procedures. The next step is to use the DOCUMENT procedure to send the output to another destination. This is called *replaying*.

Replaying Output

Program 1.4 represents the shortest, yet most useful example of PROC DOCUMENT usage. All the DOCUMENT procedure really comes down to is this basic syntax:

```

proc document name=;replay;run; quit;

```

The rest is just details. In this section, remember that all calls to the DOCUMENT procedure must include the NAME= argument.

Remember to Use RUN Statements!

Another important aspect of using the DOCUMENT procedure should be brought up before you start: The RUN statement is very important. Furthermore, like other interactive procedures such as the SQL procedure and the REG procedure, the DOCUMENT procedure must be terminated by a QUIT statement.

Program 1.4: Replaying the Output to the HTML Destination

```

ods html file="program1_4.html";
proc document name=mylib.quickstart;
  replay;
run;
quit;
ods html close;

```

6 PROC DOCUMENT by Example Using SAS

The output is shown as Output 1.1. Once your output is saved to the document, you can print it out as many times as you want, in whatever form you want.

Output 1.1: Replaying Output as HTML

The UNIVARIATE Procedure			
Variable: MSRP			
Origin=Asia Type=SUV			
Moments			
N	25	Sum Weights	25
Mean	29569	Sum Observations	739225
Std Deviation	11842.546	Variance	140245895
Skewness	1.51941545	Kurtosis	2.366336
Uncorrected SS	2.5224E10	Corrected SS	3365901476
Coeff Variation	40.050546	Std Error Mean	2368.50919

Basic Statistical Measures			
Location		Variability	
Mean	29569.00	Std Deviation	11843
Median	27560.00	Variance	140245895

Program 1.5 sends the same output as Program 1.4, but to a PDF file.

Program 1.5: Replaying the Output to a PDF File

```
ods pdf file="program1_5.pdf";
proc document name=mylib.quickstart;
  replay;
  run;
quit;
ods pdf close;
```

Output 1.2 shows the same output, but in PDF. Notice how you didn't have to re-run your original code (Program 1.1) to create a presentation of this same output in a different file format.

Output 1.2: PDF Output Excerpt

The UNIVARIATE Procedure
Variable: MSRP

Origin=Asia Type=SUV

Moments			
N	25	Sum Weights	25
Mean	29569	Sum Observations	739225
Std Deviation	11842.546	Variance	140245895
Skewness	1.51941545	Kurtosis	2.366336
Uncorrected SS	2.5224E10	Corrected SS	3365901476
Coeff Variation	40.050546	Std Error Mean	2368.50919

Basic Statistical Measures			
Location		Variability	
Mean	29569.00	Std Deviation	11843
Median	27560.00	Variance	140245895
Mode	.	Range	47637
		Interquartile Range	13391

Tests for Location: Mu0=0				
Test		Statistic	p Value	
Student's t	t	12.48422	Pr > t	<.0001
Sign	M	12.5	Pr >= M	<.0001
Signed Rank	S	162.5	Pr >= S	<.0001

You can also replay your output in a different style, as demonstrated by Program 1.6. A complete list of styles that are shipped with SAS 9.3 is given in the appendix. You can also write your own styles, and they will work with the REPLAY statement, but writing custom styles is beyond the scope of this book.

Program 1.6: Replaying the Output in a Different Style

```
ods html file="fancy.html" style=rsvp;
proc document name=mylib.quickstart;
  replay;
run;
quit;
```

With the exception of being in a different font, with different background colors, this is the same output as Program 1.5, so it is not illustrated. Printing the same output to different destinations and different styles is *the raison-d'être* of the DOCUMENT procedure.

Subsetting Output

Being able to replay your output in different formats is powerful, but the amount of output on even a medium-sized project might grow to be quite large. You need a way to replay only what you need. Program 1.7 shows how to replay only the output from the UNIVARIATE procedure. Program 1.8 demonstrates a simple version of the WHERE= clause to exclude output from your presentations. WHERE= clauses are covered in more detail in Chapter 4, “Listing Documents Using the DOCUMENT Procedure,” as well as in Chapter 5, “The REPLAY Statement.”

Output is stored in the document by each SAS procedure. Each procedure is stored in a folder named for the procedure. If there is one folder per procedure, and there is one name per folder, you might wonder if you can have more than one folder with the same procedure name in the same document. SAS will keep them separate for you through an automatic *sequence numbering* system. The characters #1 after univariate refer to the first folder named univariate. In this case, it is the only one. If there were more, you would refer to those outputs as univariate#2, univariate#3, and so on.

Program 1.7: Restricting the Replay to Include Only UNIVARIATE Procedure Output

```
ods html file="program1_7.html" style=statdoc;
proc document name=mylib.quickstart;
  replay univariate#1;
run;
quit;
ods html close;
```

Not only can you restrict your output to a particular procedure, but you can also remove specific tables from your output. In Program 1.8, the Extreme Observations tables are not displayed in the replay of the document. They were excluded because, although useful to SAS programmers, they might not be useful in a report that is presented to clients.

Subsetting output is done through WHERE= options, which work the same way as they do for the WHERE= options in the DATA step. One way among several to subset output from the DOCUMENT procedure is to use special document variables. This quick-start guide will discuss one such variable. You can search the document based on the value of `_labelpath_`, one of many special document variables. These special variables are covered in Chapter 4. The characters `!?` mean *does not contain*.

Program 1.8: Excluding Extreme Observation Output

```
ods html file="program1_8.html" style=statdoc;
proc document name=mylib.quickstart;
  replay univariate(where=(_labelpath_ !? 'Extreme Observations'));
run;
quit;
ods html close;
```


The output for Program 1.8 is the same as Output 1.7, except that the Extreme Observations tables are not present. If you created output using BY-group variables, you can also use those variables to subset output, as shown in Program 1.9. BY-group variables do not require the underscore, since they are ordinary data set variables.

Program 1.9: Replaying Output Using BY-Group Variables

```
ods html file="program1_9.html" style=statdoc;
proc document name=mylib.quickstart;
  replay univariate(where=(origin='Asia'));
run;
quit;
ods html close;
```

As with the standard WHERE= option, clauses can be combined using logical operators such as AND, OR, and NOT. Program 1.10 presents one last version of WHERE options, and brings in one more special document variable: the `_Name_` variable. Every ODS object has a name, and you will learn more about how to remember the names of your output in Chapter 2, “The ODS DOCUMENT Destination.” The two objects that are the focus of this quick-start guide are `ExtremeObs` from Program 1.8, and `Moments`, which is demonstrated in Program 1.10.

The ? operator means *contains*.

Program 1.10: Subsetting with Multiple Conditions

```
ods html file="program1_10.html" style=htmlblue;
proc document name=mylib.quickstart;
  replay univariate(where=( _labelpath_ ? 'MPG' and _name_ = 'Moments' and
    type='SUV' ));
run;
quit;
ods html close;
```

This is a good place to step back and see what you've accomplished with the DOCUMENT procedure. Output 1.3 consists of several versions of the same table, the `moments` table. Because all of the tables have the same structure, it is easy to review them as a group. It's very powerful to have the ability to save output in the SAS default order, yet replay any selection of that output based on what is important to you. This is quite a powerful feature of the DOCUMENT procedure. By using subsetting with WHERE= statements, you can now create outputs that would be quite challenging to produce without the DOCUMENT procedure. Finally, since the original output stored in the document has not changed, you can replay it in a different order next time.

Output 1.3 shows the output from the UNIVARIATE procedure, but only for `MPG_HIGHWAY` and `MPG_CITY`, and only the `moments` table.

10 PROC DOCUMENT by Example Using SAS

Output 1.3: Highway and City Gas Mileage Output

The SAS System			
The UNIVARIATE Procedure			
Variable: MPG_City (MPG (City))			
Origin=Asia Type=SUV			
Moments			
N	25	Sum Weights	25
Mean	17.32	Sum Observations	433
Std Deviation	2.76465791	Variance	7.64333333
Skewness	0.12662092	Kurtosis	-0.8227381
Uncorrected SS	7683	Corrected SS	183.44
Coeff Variation	15.9622281	Std Error Mean	0.55293158

The SAS System			
The UNIVARIATE Procedure			
Variable: MPG_Highway (MPG (Highway))			
Origin=Asia Type=SUV			
Moments			
N	25	Sum Weights	25
Mean	21.68	Sum Observations	542
Std Deviation	3.00998339	Variance	9.06
Skewness	0.16534859	Kurtosis	-0.8422186
Uncorrected SS	11968	Corrected SS	217.44
Coeff Variation	13.8836872	Std Error Mean	0.60199668

Managing Output

Although the primary function of the DOCUMENT procedure is replaying output without re-running the original analysis, another important function that it provides is the ability to manage your output. This section of the quick-start guide begins with learning a document's contents, and displaying a table of contents using the LIST statement.

Without using any options, the LIST statement displays a list of all the folders at the top level of the directory tree. This is useful for a quick overview of the ODS document's contents.

Program 1.11: Creating a Summary Table of Contents

```
proc document name=mylib.quickstart
  list; /* one line for each proc */
run;
quit
```

In Output 1.4 there is a short listing of the document, with one line for each procedure stored. The sequence numbers are placed automatically.

Output 1.4: High Level Overview of the Document

Listing of: \Mylib.Quickstart\		
Order by: Insertion		
Number of levels: 1		
Obs	Path	Type
1	Contents#1	Dir
2	Univariate#1	Dir
3	Means#1	Dir
4	Freq#1	Dir

The ODS document item store actually contains a complete directory structure, and there are multiple levels of directories in a typical one. To see the entire directory tree, use the LEVELS=ALL option, which shows all levels down to the procedure output level. The DETAILS option shows additional fields, such as when the output was created. For more information about the fields shown in the DETAILS statement, see Chapter 4, “Listing the Documents Using the DOCUMENT Procedure” as well as Chapter 10, “Working with Links” and Chapter 11, “Working with Templates.”

Program 1.12: Creating a More Detailed Listing

```
proc document name=mylib.quickstart;
  list / levels=all details;/* every subdirectory */
run;
quit;
```

Output 1.5 shows the results. Here are a few features worth noting. The listing shows every element that is stored in the document. The actual output that you would see if you replayed the document is listed as type TABLE. TABLE is one of several types of *output objects* that can be stored in the document. Each entry for a table also shows its size and its full path. The list also shows every directory and subdirectory, along with a descriptive label. If you look at the full path for an output object, such as observation #10, the last entry in the path is the *ODS name* of the object. You can see the familiar moments object along with the ExtremeObs that were featured in Program 1.10 and Program 1.8, respectively.

12 PROC DOCUMENT by Example Using SAS

Output 1.5: The Detailed Listing with All Levels Shown

Listing of: \Mylib.Quickstart\				
Order by: Insertion				
Number of levels: All				
Obs	Path	Type	Size in Bytes	Label
1	\Contents#1	Dir		The Contents Procedure
2	\Contents#1\DataSet#1	Dir		WORK.CARS
3	\Contents#1\DataSet#1\Attributes#1	Table	716	Attributes
4	\Contents#1\DataSet#1\EngineHost#1	Table	563	Engine/Host Information
5	\Contents#1\DataSet#1\Variables#1	Table	557	Variables
6	\Contents#1\DataSet#1\Sortedby#1	Table	309	Sortedby
7	\Univariate#1	Dir		The Univariate Procedure
8	\Univariate#1\ByGroup#1	Dir		Origin=Asia Type=SUV
9	\Univariate#1\ByGroup#1\MSRP#1	Dir		MSRP
10	\Univariate#1\ByGroup#1\MSRP#1\Moments#1	Table	613	Moments
11	\Univariate#1\ByGroup#1\MSRP#1\BasicMeasures#1	Table	522	Basic Measures of Location and Variability
12	\Univariate#1\ByGroup#1\MSRP#1\TestsForLocation#1	Table	574	Tests For Location
13	\Univariate#1\ByGroup#1\MSRP#1\Quantiles#1	Table	597	Quantiles
14	\Univariate#1\ByGroup#1\MSRP#1\ExtremeObs#1	Table	408	Extreme Observations

Changing Titles and Footnotes

If you notice in the detailed listing, you see directories and output objects. Output objects include TABLE and GRAPH objects, although there are other kinds as well. Everything that is discussed for these two types of output carries over to the others. For the record, there are also the following types of output that can be stored in a document.

- REPORT
- CROSSTAB
- EQUATION

Output objects have their own titles and footnotes that can be changed with the OBTITLE and OBFOOTN statement. Program 1.13 shows how to change a title. Once changed, that title stays stored with the document until it is changed again. When you replay, the title that is stored with the object usually takes precedence over the active SAS title. (This behavior can be overruled. This is covered in Chapter 5, “The REPLAY Statement.” See the discussion on the ACTIVETITLE option.)

Program 1.13: Changing a Title and Replaying It

```

proc document name=mylib.quickstart;
  obtitle \Univariate#1\ByGroup1#1\MSRP#1\Moments#1 'Manufacturer''s
          Suggested Retail Price';
  replay \Univariate#1\ByGroup1#1\MSRP#1\Moments#1;
run;
quit;

```

In this example, in order to get the object name into the statement, the author cut and pasted the pathname directly from the output window. Seeing how output objects can be rather long, it won't surprise you to know that there are easier and shorter ways to specify output objects. However, cutting and pasting is reasonable if you are changing only one or two output objects at a time. Some of these other methods will be discussed in Chapter 6, "Managing Folders" and Chapter 7, "Customizing Output." Output 1.6 shows the output with the new title in place.

Output 1.6: Changing a Title of an Output Object

Moments			
N	25	Sum Weights	25
Mean	20560	Sum Observations	730205

Using Folders

You can create your own folders and subfolders using the MAKE statement, and then instruct the DOCUMENT destination to save your output to your new directory. Program 1.14 shows the creation of a new directory, and saving output to it.

Program 1.14: Saving Output to a User-created Directory

```

proc document name=mylib.quickstart;
  make summaries;
run;
quit;

ods document name=mylib.quickstart dir=(path=\summaries);
proc report nowd data=cars;
  column origin type msrp;
  define origin / group order=data;
  define type / 'Veh. Type' group order=data;

```

14 PROC DOCUMENT by Example Using SAS

```
define msrp / analysis mean;
run;
ods document close;
/* now open the document for viewing */
proc document name=mylib.quickstart;
  list summaries; /* omitting sequence no defaults to highest */
run;
quit;
```

Output 1.7 shows how the document now appears, with the summaries#1 directory and the output for the REPORT procedure inside that. There are three levels to this directory, and the actual table is at the third level. Sometimes you need to look carefully at the document produced so that you can find your output. It might be structured differently than you expect at first.

Output 1.7: Showing Folder for REPORT Procedure Nested in Summaries#1

Listing of: \Mylib.Quickstart\summaries#1				
Order by: Insertion				
Number of levels: All				
Obs	Path	Type	Size in Bytes	Label
1	\summaries#1\Report#1	Dir		The Report Procedure
2	\summaries#1\Report#1\Report#1	Dir		Detailed and/or summarized report
3	\summaries#1\Report#1\Report#1\Report#1	Table	1406	

If you want to replay the report, and only the report, any of the following three commands shown in Program 1.15 would help. Program 1.15 also features several statements used in the same DOCUMENT procedure call. Each is separated by a RUN statement. The second REPLAY statement omits the sequence number #1. If omitted, the sequence number will default to the most recent version of the output.

Program 1.15 : Replaying the Report

```
proc document name=mylib.quickstart;
  replay summaries#1;
run;
replay Summaries#1\Report; /* Report becomes Report#1 */
run;
replay Summaries#1\Report#1\Report#1;
run;
quit;
```

When you are searching on the *value* of the path, the search is case sensitive. However, the pathname itself is case insensitive.

Thus, the following statement will still replay correctly.

```
replay summaries#1\report#1\report#1;
run;
```

By contrast, when you compare the `_PATH_` variable to a string constant, as in Program 1.10, the string must match exactly, including case.

Deleting and Overwriting Document Contents

Program 1.16 shows how to clear an entire document. The new idea in this code is the *WRITE access mode*. The default behavior of the document destination is to open documents in UPDATE access mode. This causes new output to be placed into the document after any existing output. The WRITE access mode opens a document and clears its contents before writing.

Program 1.16: Starting Over

```
ods document name=mylib.quickstart(write);
<different, newer sas code >
ods document close;
```

The document `mylib.quickstart` will now have only the results of the body of the new SAS code. The original analysis from Program 1.1 will no longer be retrievable.

Program 1.17 demonstrates how to delete documents with the DOCUMENT procedure. The DELETE statement does not give you any second chances, so be sure you want to delete before you do so. Nonetheless, sometimes a document can grow to become unwieldy, and deleting and starting over might be the best choice, as in Program 1.16. Still, there are usually better alternatives to deleting your hard work, and Chapter 6, “Managing Folders” discusses some of these alternatives, such as archiving unwanted work in another document or folder.

When invoking the DOCUMENT procedure, you must have a current document, even when you are using the DELETE command to delete the current document, as in Program 1.17.

Program 1.17: Deleting a Document

```
proc document name=dontdoit;
  delete dontdoit;
run;
quit;
```

What Remains to Be Covered?

Now that you have learned to create a document, store output, and replay the output, you might be wondering what is left to learn. The remaining chapters will show you how to refine the basic skills presented in the quick-start guide, such as:

- Learning more ways to conditionally subset when replaying part of a document
- Using the MOVE TO statement with positioning options to change the order in which output is to be replayed
- Storing more than one procedure of the same name in a document
- Labeling your output so that you know the purpose for a collection of output as well as its contents
- Changing titles and footnotes before replaying the document
- Adding text notes to your work that appears when you replay
- Making many modifications by changing the ODS Template that was used to create the original output.

Summary

The ODS Document Facility maintains an ongoing journal of your output that you can replay at any time without re-running the original analysis. This approach has several advantages. By delaying the tasks of printing and changing layout and appearance, you can focus on your main work. When it comes time to present the output, you can experiment with a variety of destinations, output formats, and style options, all without making any changes to or re-running your original code.

About The Author



Michael Tuchman is a senior statistician at Accolade, Inc., a leading provider of Professional Health Assistant services. A SAS user since 1999, Tuchman applies his extensive knowledge of SAS to understanding how customers utilize healthcare, ultimately empowering them to get more value out of the healthcare system. Prior to joining Accolade, he worked at SDI Health (now part of IMS Health) where he mined healthcare data to find patient behavior patterns. Prior to joining SDI, he developed predictive models in the Health Informatics Department at Aetna. Tuchman received a BS from the Massachusetts Institute of Technology, and an MS in mathematics from the University of North Carolina at Chapel Hill.

Learn more about this author by visiting his author page at support.sas.com/tuchman. There you can download free chapters, access example code and data, read the latest reviews, get updates, and more.

ACCELERATE YOUR SAS[®] KNOWLEDGE WITH SAS BOOKS.

Visit the [SAS[®] Press author pages](#) to learn about our authors and their books, download free chapters, access example code and data, and more.

Browse our [full catalog](#) to find additional books that are just right for you.

Subscribe to our [monthly e-newsletter](#) to get the latest on new books, documentation, and tips—delivered to you.

Browse and search [free SAS documentation](#) sorted by release and by product.

Email us: sasbook@sas.com
Call: 800-727-3228

