

Paper 172-2008

Steps to Success with the SAS® Output Delivery System

Andrew H. Karp

Sierra Information Services

Sonoma, CA USA

ABSTRACT

Implementation of the Output Delivery System (ODS) has dramatically changed how SAS users can work with their output. This paper provides an overview of how ODS “works” within the SAS System and how you can use it to take your SAS output and “deliver” it to “destinations” such as a PDF, RTF or HTML file. This paper is designed to equip newer users of SAS (or experienced SAS users new to the Output Delivery System) to get started apply ODS functionalities in their reports and projects. After mastering the concepts and techniques discussed in this paper you will be able, with just a few extra commands in your programs, to take your POLDO (Plain Old Listing Destination Output) and convert in to visually-appealing reports without a lot of manual cutting-and-pasting or manual re-entry of SAS-generated output in to other file formats. And, you will have the insights necessary to explore more advanced ODS capabilities that are discussed in many of the references provided at the end of the paper.

INTRODUCTION

The SAS Output Delivery System was added to SAS System Software in Version 7, which was not around for very long until SAS released Version 8 of its product. So, most long-time SAS users, including the author, started applying ODS capabilities with the release of SAS Version 8 software, since most of us never used V7. ODS is an “embedded” part of the SAS System, it is not an “extra” or “add-in” product. Instead, it is the part of BASE SAS Software that controls the “delivery” of SAS output to “destinations.” Since the default “destination” is LISTING, or your Output Window, many SAS users are often surprised to learn that they are “using ODS” every time they run a procedure step whose results are displayed in their Output Window. So, you probably have more experience using ODS than you thought you did before you starting reading this paper!

STEP 1: UNDERSTANDING HOW ODS “WORKS”

The first step to successfully applying ODS tools to your work is to understand what it is doing “under the hood” as you write and submit SAS programs for execution. In a nutshell, when you execute an “ODS compliant” (see below) SAS Procedure Step, the PROC generates one or more Data Components and automatically passes them off to the Output Delivery System, which takes things from there. ODS binds the PROC-generated Data Component to a Table Template and a Style Template, both of which are explained in more detail below. The result is a “object” or “table” (both terms are used interchangeably) which ODS then “delivers” to all Destinations that are currently “open” in your SAS Session. As we will see later on in this paper, there are many Destinations, and many Style Templates to which we can deliver our output. And, we can even create customized Table and Style Templates if we want, using PROC TEMPLATE, which is part of the BASE SAS module.

A Table Template contains the instructions ODS uses to take the contents of the Data Component and structure it in to the rows and columns you see in the output. Most SAS users seem to be happy with the default Table Templates that are part of ODS; how to create customized versions are far beyond the scope of this paper. Please see the ODS Documentation manuals, the ODS “community” page at support.sas.com, or call SAS Technical Support for help if you really need to create a customized Style Template.

Style Templates contain the information ODS uses to assign appearance or “style” attributes to your SAS-generated output. Features such as fonts, colors, shading, background colors, font sizes, etc. are “controlled” by Style Templates. As of SAS 9.1.3 Software there are 43 “internal” or SAS-supplied Style Templates from which to choose. We’ll talk a bit more about Style Templates later in this paper, but won’t have the time to address how to create your own customized Style Templates, which is possible using PROC TEMPLATE in the BASE SAS Module. For an excellent introduction to creating customized Style Templates, please read Lauren Haworth’s paper “SAS With Style: Creating Your Own ODS Style Template.” A link to a PDF version of Lauren’s paper is in the “References” section at the bottom of this paper.

DO ALL SAS PROCS “WORK” WITH ODS?

No. It’s important to remember that ODS works with Procedure-generate output, such as the tables, statistics and other information that we typically see in our Output Window (i.e., the LISTING Destination) Some SAS PROCS

“only” create a SAS data set. For example, PROCs FORMAT, SORT and TRANSPOSE, all in the BASE SAS Module do not “communicate” with ODS, as they do not create “data components” which are then passed on to the ODS for additional processing before “delivery” to destinations.

HOW DOES A PROC'S “NOPRINT” OPTION IMPACT ITS ROLE WITH THE ODS?

Let's address this critical topic right away, since it comes up often, especially with SAS users who code a NOPRINT option in their Procedure steps to have the PROC store its “work” in either a permanent or temporary SAS data set. There are a number of PROCs in the BASE, STAT and ETS modules that support a NOPRINT option. Regardless of the specific Procedure where you code a NOPRINT option, that option disables the PROCs ability to send Data Components to the ODS for subsequent delivery to a “Destination.” We will talk about this more in the context of the ODS OUTPUT Destination later in this paper.

STEP 2: MIND YOUR TABLE (MANNERS)

As discussed above, ODS-compliant PROCs send the results of their work to ODS in a series of Data Components, which are then used to create Output Objects or Tables that are delivered to Destinations. Now that we understand the basics of how ODS operates, the next step to success with ODS is learning the names of the tables our Procedure Steps are generating. With that information in hand we can then control which tables are generated and the destinations to which they are delivered.

Here are two pieces of information to keep in mind about ODS table names: first, the name of the table (e.g. FitStatistics) corresponds to the contents of the table, making it easier for you to remember what's in which tables your Procedure Steps generate. Second, table names are not necessarily unique to one SAS PROC. For example, many of the statistical modeling procedures in SAS/STAT Software can generate a table called “ParameterEstimates.”

We can learn the ODS Table Names for the PROCs with which we work in one of two ways: looking in that PROCs documentation manual chapter, or by using the ODS TRACE ON command prior to running the PROC.

LEARNING ODS TABLE NAMES IN THE SAS DOCUMENTATION MANUALS

All ODS-compliant SAS procedures have an “ODS Table Names” sub-section in the “Details” section of their documentation chapter. For example, here is what a piece of the “ODS Table Names” sub-section of the PROC FREQ documentation looks like:

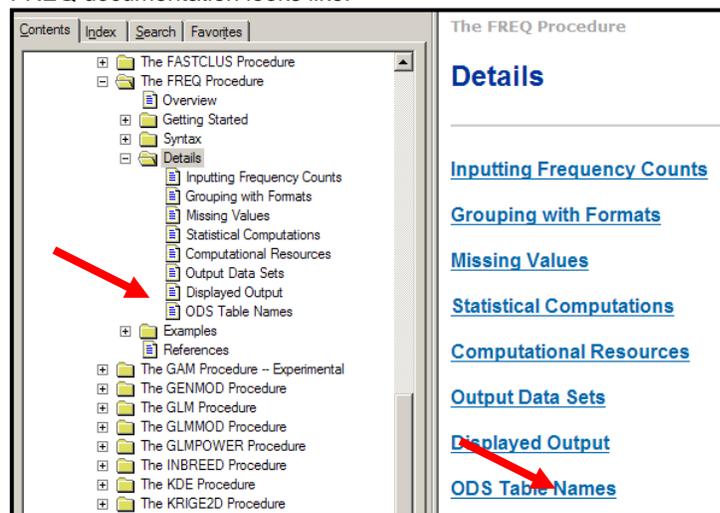


Figure 1: PROC FREQ On-Line Documentation (SAS 9.1.3)

The FREQ Procedure

ODS Table Names

PROC FREQ assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 29.11 lists the ODS table names together with their descriptions and the options required to produce the tables. Note that the ALL option in the TABLES statement invokes the CHISQ, MEASURES, and CMH options.

Table 29.11: ODS Tables Produced in PROC FREQ

ODS Table Name	Description	Statement	Option
BinomialProp	Binomial proportion	TABLES	BINOMIAL (one-way tables)
BinomialPropTest	Binomial proportion test	TABLES	BINOMIAL (one-way tables)
BreslowDayTest	Breslow-Day test	TABLES	CMH ($h \times 2 \times 2$ tables)
CMH	Cochran-Mantel-Haenszel statistics	TABLES	CMH
ChiSq	Chi-square tests	TABLES	CHISQ
CochransQ	Cochran's Q	TABLES	AGREE ($h \times 2 \times 2$ tables)
ColScores	Column scores	TABLES	SCOROUT
CommonOddsRatioCL	Exact confidence limits for the common odds ratio	EXACT	COMOR
CommonOddsRatioTest	Common odds ratio exact test	EXACT	COMOR
CommonRelRisks	Common relative risks	TABLES	CMH ($h \times 2 \times 2$ tables)
CrossList	Column format	TABLES	CROSSLIST

Figure 2: ODS Table Names Information (Excerpt) from the SAS 9.1.3 PROC FREQ Documentation Chapter

LEARNING ODS TABLE NAMES USING THE ODS TRACE COMMAND

The second approach to learning ODS Table names is to utilize the ODS TRACE command prior to running a Procedure Step. This command instructs SAS to write the names of the tables generated during Procedure Step execution to either the SASLOG (the default), or if you add the LISTING option to the command, in to the Output Window just before the table itself. Most people seem to like having the ODS TRACE command-generated information in the Output Window, since it makes it easier to understand what output is in which table. So, here is an example of what happens when we code an ODS TRACE ON/LISTING command prior to running a PROC FREQ Step. The code is:

```
ODS TRACE ON/LISTING;
proc freq data=lung. lung_anon(where=(mom_smk ne 9));
format sex sexfmt. mom_smk ynfmt.;
tables Sex*mom_smk/CHI SQ EXPECTED;
title3 'Cross Tab with Chi-Square, Expected Options';
title4 'Learning Table Names with ODS TRACE ON/LISTING';
run;
ODS TRACE OFF;
```

Notice that after the PROC FREQ step I have coded the ODS TRACE OFF command. That command instructs SAS to stop writing table name information in to either the SASLOG or the LISTING destination, depending on whether or not you added the LISTING option to your ODS TRACE ON statement. It's a good idea to submit an ODS TRACE OFF command after running the Procedure Step from which you want the ODS table name information. Otherwise, the ODS TRACE ON command will remain in effect for the rest of your SAS session and you'll probably wind up with a lot of information in either your SASLOG or Output Window you really don't need.

Here is what the Output Window appears after running the ODS and PROC FREQ statements above:

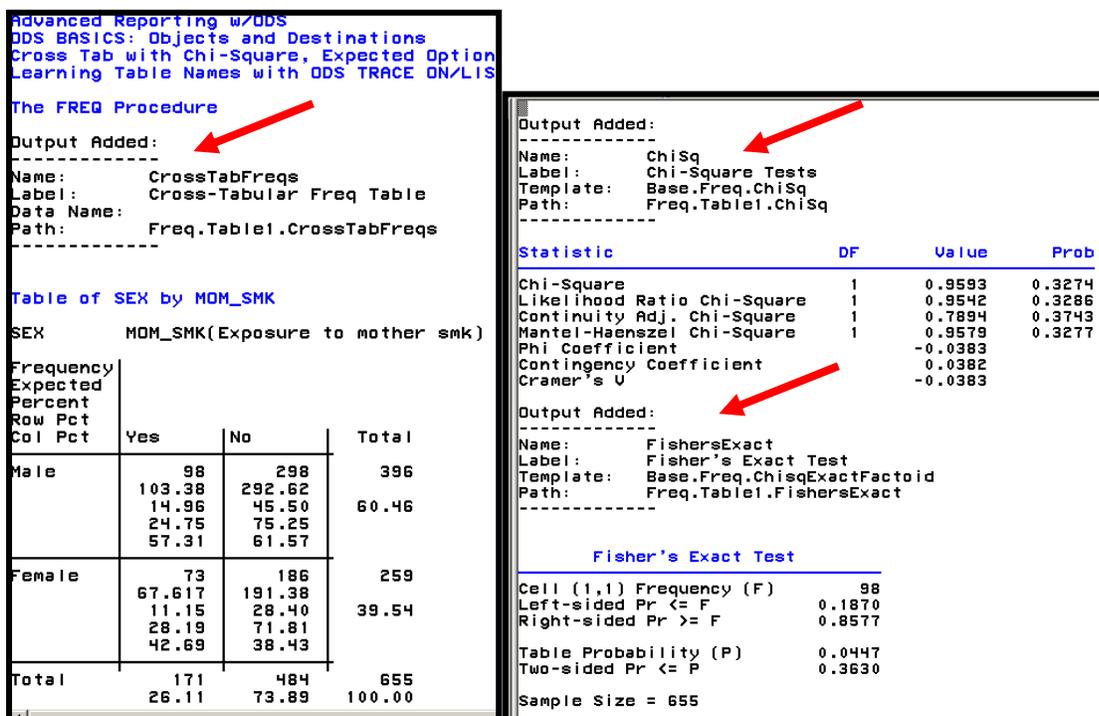


Figure 3: Output Window with Results from ODS TRACE ON/LISTING Command

Figure 3 shows both the PROC FREQ-generated results and the ODS Table Names which contain that output. So, we can see that, for example, the Chi-Square-based tests are in the CHISQ Table and the number of cases in each cell (or "box") in the requested cross-tabulation are in the CROSSTABFREQS table.

One other thing to keep in mind as you start working with Procedure generated tables is that each Procedure has a different set of default and optional tables that it can generate. That's why its important to become familiar with the tables generated by the Procedures you tend to use often, and to further understand what values/statistics are in those tables.

And, the contents of those tables are also controlled by a series of Procedure-specific defaults and options. In this example, we obtained the CHISQ and FISHERSEXACT tables because the CHISQ option was specified in PROC FREQ's TABLES Statement. Coding the EXPECTED option in that statement meant that PROC FREQ computed the expected number of cases in each cell (under the statistical null hypothesis of independence of the row and column variables) and added them to the default contents of the CHISQ table.

Another way to see information about the ODS tables generated by your Procedure Step is to examine what's in the Results Window, which is typically on the left-hand-side of your SAS "desktop." Here is a screen capture of both my Results Window after running the PROC FREQ step which generated the output displayed in Figure 3. The icons immediately to the right of the table labels indicates that the contents of those tables have been output to the LISTING Destination (i.e., the Output Window).

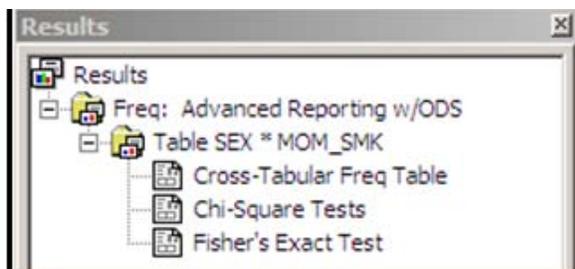


Figure 4: Contents of Results Window Showing ODS Table Name Information

STEP 3: TAKE WHAT YOU NEED AND LEAVE THE REST

Now that we understand the basics of how ODS works and also know how to find out the names of ODS-generated objects (or tables), we can now explore how to control what tables/objects ODS will generate for us during a Procedure Step “run.”

This ODS feature is especially important for those of you who run many PROC steps in a single project or program and who only some of the (at times) many times that a PROC will generate by default. For example, someone using PROC LOGISTIC to generate dozens different predictive models may not need all of the eight tables that that procedure generates by default for each model they are testing. Instead, they can “prune” the amount of output appearing in their Output Window (or any other ODS destination—see below) to just the tables they need to move forward in their project. The might just want to see the model fit statistics in the FitStatistics table and the model’s parameter estimates in the ParameterEstimates table, and not have to wade through the rest of the default PROC LOGISTIC-generated output.

To do this, we need to understand the ODS SELECT and EXCLUDE statements. These statements are placed before your PROC step and give ODS instructions on which tables you want SELECTed or EXCLUDed from your output. Just like the KEEP and DROP SAS Programming Language and Data Set Options, you can use either a SELECT or an EXCLUDE, depending on what’s easiest for you. (But, keep in mind that KEEP and DROP are used to control what variables are in a SAS data set while the ODS SELECT and EXCLUDE statements control what ODS tables are generated. And, ODS SELECT is NOT the same thing as the SAS Programming SELECT statement.

Returning to my PROC FREQ-generated cross-tabulation, suppose I really like the contents of the CHISQ table, but don’t have any need for what’s in the FISHERSEXACT table. But, by default, both tables are generated when I put a CHISQ option in my TABLES Statement in a PROC FREQ step. In order to remove the FISHERSEXACT table from my output, I can use either an ODS SELECT Statement to select the tables I want or an ODS EXCLUDE statement to exclude the tables I don’t want. For completeness, the code below shows **both** an ODS SELECT and an ODS EXCLUDE, but the EXCLUDE Statement has been “commented out,” since we only need one of them to get what we need.

```
ODS SELECT CrossTabFreqs Chi SQ;
* ODS EXCLUDE FishersExact;
proc freq data=lung.lung_anon(where=(mom_smk ne 9));
format sex sexfmt. mom_smk ynfmt.;
tables Sex*mom_smk/CHI SQ EXPECTED;
title3 'Cross Tab with Chi-Square, Expected Options';
title4 'Selectng Objects';
run ;
```

In this example, PROC FREQ and ODS will only generate the two tables SELECTed by the ODS SELECT statement. Since we did not SELECT the FishersExact table, it won’t be generated and “delivered” to the LISTING Destination. We can see this by again looking at the Results Window displayed in Figure 5 below.

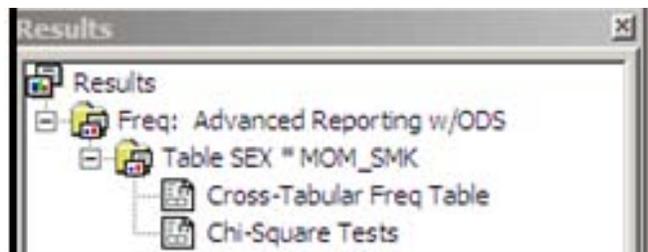


Figure 5: Results when Using an ODS SELECT Statement.

By default, your table selection or exclusion lists are re-set back to the default ALL at either an implicit or explicit “step boundary.” For those of you not familiar with that term, an “explicit step boundary” is either a RUN or QUIT at the end of a PROC (or DATA) step and an “implicit step boundary” is a PROC or DATA statement that follows a previous PROC or DATA Step. If, for example, you have two PROC Steps in your program, and there’s no explicit step

boundary between them, the second PROC statement serves as the “implied step boundary” between it and the preceding Step. (As an aside, the absence of an explicit RUN or QUIT statement for the last DATA or PROC Step in a programming running under SAS in a Windows-based operating system means that the last “step” will run “indefinitely,” or until you supply an explicit step boundary to SAS to terminate that step.)

STEP 4: MAKE THE OUTPUT DESTINATION WORK FOR YOU

So far in this paper we’ve learned how ODS works within the overall framework of SAS System software, how to learn the names of the tables our SAS Procedures generate, and how to select or exclude some tables from being generated when we run an ODS-compliant SAS Procedure. But, all of the output we’ve created so far has been delivered to the Output Window, or LISTING Destination. In this Step to Success with ODS we will look at the OUTPUT Destination, to which we can deliver ODS-generated tables as either permanent or temporary SAS Data Sets.

The OUTPUT Destination, in my opinion, deserves a separate “step” in this paper because it “behaves” differently than what I like to call “presentation” destinations such as PDF, RTF or HTML, which are discussed below. With “presentation” destinations we use a combination of an ODS Style Template and, perhaps, destination-specific options to take our SAS-generated output and make it “pretty” by controlling its appearance. More on this later.

First, let’s explore what the OUTPUT Destination can do for us, and how it is a bit different than other destinations.

The OUTPUT destination creates either permanent SAS Data Sets from an ODS-generated table. These data sets are just like data sets created by other processes within the SAS System: they have both a descriptor portion and a data portion. We can work with them in Data Steps just like Data Sets and apply procedure steps to them.

One of the many benefits to the OUTPUT Destination is it gives you the ability to “gather” the results from multiple Procedure Step runs in to one data set that you can then display using other SAS tools. This is often something people who use SAS’ analytic procedures for statistical modeling really want to do. They may need to run the same analytic PROC many times to generate different statistics and then “pick” just a few of the statistics the PROCs generate in to a customized table for presentation in a paper or report.

Before ODS came along, this process often involved a lot of tedious and time consuming manual labor to wade through pages of SAS analytic procedure output and then cut-and-paste some of the parts in to some other table format. Not only was this very boring, but it was also quite prone to error as data were manually transferred from, say, the SAS Output Window (or hardcopy printouts) to a spreadsheet file. And, if new data became available or other aspects of the research project required adding new models to the project, these manual steps needed to be repeated over and over again.

Now, with the ODS Output Destination we can eliminate most, if not all, of the manual labor required to generate a customized data set containing just the pieces of Procedure-generated output we want.

This example shows how the Output Destination is used to take what’s in the CHISQ table generated by a PROC FREQ step and save it in a permanent SAS data set. That data set’s data portion is then displayed using a PROC PRINT step. This example also demonstrates another important ODS capability: delivering the same output table to multiple destinations at a time. First, here’s the SAS code to generate the output displayed in Figure 7 below:

```
ods output chi sq = lung.chi sq_1;
proc freq data=lung.lung_anon(where=(mom_smk ne 9));
format sex sexfmt. mom_smk ynfmt.;
tables Sex*mom_smk/CHI SQ EXPECTED;
title3 'Using the OUTPUT Destination';
run;

proc print data=lung.chi sq_1;
title3 'Data Set from the CHI SQ Table';
run;
```

In this example, we are creating permanent SAS Data Set CHISQ_1 which is stored in the LUNG library (to which a library reference had been previously established using either the New Library window or the LIBNAME programming statement. So, PROC FREQ does the “work” requested in the TABLES statement and delivers a total of three Data Components to ODS, which we’ve seen earlier: CrossTabFreqs, ChiSq and FishersExact. All three will be “delivered”

to the LISTING Destination, and the contents of the ChiSq table will also be delivered to the specified permanent SAS data set LUNG.CHISQ_1. This is shown in Figure 6 below, which displays the Results Window after executing the PROC FREQ and PRINT steps:

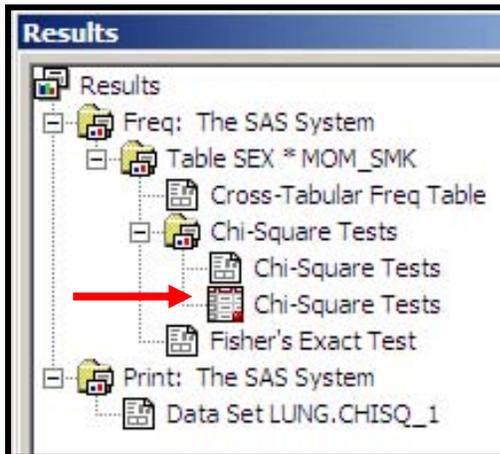


Figure 6: Results Window with OUTPUT Destination Icon

Looking at Figure 6 we see a “SAS Data Set” icon and a “Listing Destination” icon listed under the Chi-Square Tests label. That tells us that the contents of the Chi-Square table have been “delivered” by ODS to two separate destinations. In a subsequent example we will learn how to “open” and “close” ODS destinations at various points in our program so that we get just what we want where we want it. But, it’s important to understand that ODS delivers output objects to as many destinations as are open at that point in your program. In this example, the LISTING destination was open (by default) for all three tables generated by the PROC FREQ step and the OUTPUT destination was open for the ChiSQ (Chi-Square) table. Table 7 shows the PROC PRINT-generated output delivered by ODS to the LISTING Destination of the Data Set created by the above PROC FREQ and ODS statements:

Obs	Table	Statistic	DF	Value	Prob
1	Table SEX * MOM_SMK	Chi-Square	1	0.9593	0.3274
2	Table SEX * MOM_SMK	Likelihood Ratio Chi-Square	1	0.9542	0.3286
3	Table SEX * MOM_SMK	Continuity Adj. Chi-Square	1	0.7894	0.3743
4	Table SEX * MOM_SMK	Mantel-Haenszel Chi-Square	1	0.9579	0.3277
5	Table SEX * MOM_SMK	Phi Coefficient	—	-0.0383	—
6	Table SEX * MOM_SMK	Contingency Coefficient	—	0.0382	—
7	Table SEX * MOM_SMK	Cramer's V	—	-0.0383	—

Figure 7: SAS Data Set Created by OUTPUT Destination Displayed Using PROC PRINT

This example demonstrates the relative ease with which we can now “capture” the statistics generated by SAS analytic procedures in SAS data sets for further analysis or reporting. With the ODS Output Destination we can usually eliminate “PROC CUT-and-PASTE” and/or “PROC HIGHLIGHTER” steps to manually create the tables we need for our final reports and analyses.

Here’s another example of using the ODS OUTPUT Destination. In the examples we’ve seen so far in this paper PROC FREQ’s analytic capabilities were used to assess the statistical independence between a lung cancer patient’s gender (SEX) and their mother’s smoking history (MOM_SMK). The researcher is not only interested in the relationship between those two variables, but between the patient’s gender and two other measures: father’s smoking status (POP_SMK) and spouse’s status (SPC_SMK). What the researcher wants is a SAS data set that contains just the first statistic (Chi-Square) and its associated probability value (or p-value) for each of these analyses in one table to then display in their report.

By taking advantage of a combination of ODS OUTPUT Destination tools and SAS Data Set Options the desired output is generated in one PROC FREQ Step preceded by a single ODS OUTPUT Statement:

```
ods listing close;
ods output chisq = lung.chisq_3(where=(upcase(Statistic)= 'CHI-SQUARE' ));
proc freq data=lung.lung_anon(where=(mom_smk ne 9 and pop_smk ne 9 and
                                spc_smk ne 9));
format sex sexfmt. mom_smk pop_smk spc_smk ynfmt. ;
tables Sex*(mom_smk pop_smk spc_smk)/CHISQ ;
title3 'Using the WHERE Clause Data Set Option in the OUTPUT Destination';
run;
ods listing;
```

This code above shows that three separate cross-tabulations are requested in the TABLES statement. Since the ODS LISTING CLOSE instructs ODS to only create the data set (LUNG.CHISQ_3) specified in the OUTPUT Statement. At the end of the PROC FREQ Step the ODS LISTING command is issued to re-open that Destination for output from future “runs” of ODS-compliant SAS procedures.

The WHERE Clause SAS Data Set Option is used to select the rows/observations which will be output by ODS to LUNG.CHISQ_3. Using the “equals-colon” or “starts with” modifier in the WHERE Clause instructs ODS to only output to LUNG.CHISQ_3 those observations whose values of the variable STATISTIC (translated to all upper-case letters with the UPCASE SAS Programming Language Function) “starts with” “CHI-SQUARE.”

The resulting data set is displayed in Figure 8 below, and it demonstrates the relative ease with which we can now get the pieces of SAS Procedure-generated output we need in to one table by taking combining the ODS LISTING procedure output please download a copy of another one of my user group presentations, ODS 201: Making Your Life Easier with the SAS Output Delivery System,” from the “User Group Presentations” link at www.sierrainformation.com

Here is the PROC PRINT display of Data Set LUNG.CHISQ_3:

**Advanced Reporting with ODS
Objects, Tables and Destinations
Using the WHERE Clause Data Set Option in the OUTPUT Destination**

Data Set: lung.chisq_3

<u>Obs</u>	<u>Table</u>	<u>Statistic</u>	<u>DF</u>	<u>Value</u>	<u>Prob</u>
1	Table SEX * MOM_SMK	Chi-Square	1	0.7422	0.3890
2	Table SEX * POP_SMK	Chi-Square	1	0.4533	0.5007
3	Table SEX * SPC_SMK	Chi-Square	1	43.5712	<.0001

Figure 8: Data Set Created by the ODS OUTPUT Statement

Before moving on to the next section of this paper, which deals with some ODS’ “presentation” destinations, I’d like to address a question that I am often asked when I talk about the ODS Output Destination at SAS user group events. That question is usually phrased along the lines of “what is the difference between the SAS data sets created by the ODS OUTPUT destination and those which can be generated using, for example, the OUTEST option in a SAS analytic procedure and/or the OUTPUT Statement, which is available in PROCs MEANS, SUMMARY and many analytic PROCs in the STAT Module?”

There are important differences between how these data sets are created, what they contain and the purposes to which they might be used. First, data sets created by the OUTEST and OUTPUT Statements in a Procedure Step are created by that PROC, and not by ODS. These are separate processes within the SAS System. Second, while the data sets created by the OUTEST option often have the same statistics/values as those created when you use the ODS OUTPUT Destination, the contents and structure of the data sets are often dramatically different. One major reason for this difference is that data sets created by the OUTEST option in a SAS analytic procedure are designed to be used in conjunction with other SAS PROCs, such as PROC SCORE, and were not designed to necessarily meet reporting or data display requirements.

Data sets using the OUTPUT Statement in analytic procedures such as PROC REG or PROC LOGISTIC (just to name two that support this statement) generate a SAS data set that contain as many observations in the output data

set as there are in the input data set, and depending on how you write your OUTPUT statement, that data set may contain statistics (such as a test of whether each individual observation is statistically “influential”) which are generated by the PROC. When using the OUTPUT Statement in PROC MEANS or SUMMARY you can create data sets containing statistics computed on the “analysis variables” you code in your Procedure Step. But, keep in mind that these data sets are created by the Procedure, not by the Output Delivery System. For more insights in to the OUTPUT Statement and how to use it in PROC MEANS (and SUMMARY), please download copies of my presentations “Steps to Success with PROC MEANS” and “Getting the Most from PROC MEANS” from my website at www.sierrainformation.com.

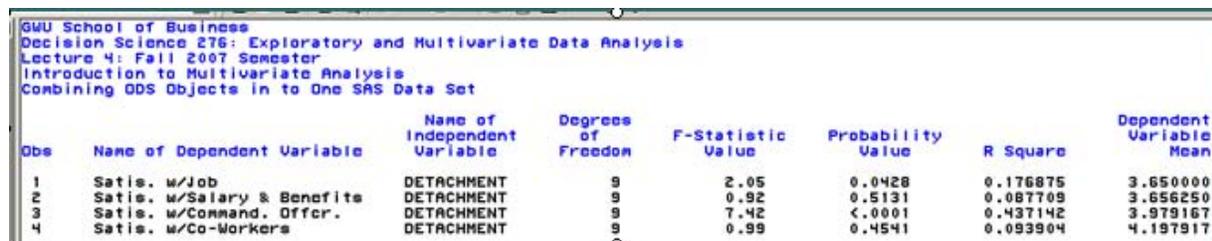
STEP 5: KNOW YOUR DESTINATION

In the previous Step to Success we looked at several examples of why we might want to use the OUTPUT Destination to save Procedure-generated output in a SAS data set. When we want to “present” our output in a report, however, it makes more sense to use one of what I like to call the “presentation” destinations, such as:

- **PDF:** Delivers results to a Portable Document File. Perhaps the most popular way of sharing SAS-generated reports and analyses, this destination creates a PDF that you can share with people who don't have SAS Software. The PDF “drivers” are built in to ODS, so you don't need Adobe Acrobat™ or other third-party software to create them using this destination. Once you put your SAS output in a PDF you can easily email it to the recipient or add it to a web site. All the recipient needs is the free Adobe Reader to view the results. And, it is VERY difficult for someone to alter the contents of a PDF, so you usually don't have to worry about someone “changing” the results of your analyses!
- **HTML:** Delivers results to a Hypertext Markup Language file. This destination allows you to save SAS Procedure-generated output as “web pages” that you can then display on a web site. The HTML destination makes it very easy to share your output via an internet or intranet site. The more you know about HTML (about which, frankly, this author knows very little), the easier it is for you to use this destination to accomplish your reporting projects. Using the HTML destination allows you to create “clickable” Tables of Contents and/or Tables of Pages, as well as link reports together to “drill down” from one level of analysis to another.
- **RTF:** Creates a Rich Text File (RTF) from your SAS-generated output, which is particularly useful if you want to take your SAS Procedure output and introduce in to an MS Word or Powerpoint file. Unfortunately, we don't have “Word” or “Powerpoint” Destinations in ODS, so the RTF Destination is often a useful “bridge” between the output SAS can create for you and your final MS Word report or Powerpoint presentation.

These are just three of several dozen presentations available in SAS 9.1.3 Software. Others, which are not covered in this paper, include the XML, LATEX, PRINTER, DOCUMENT, CSV and CSVALL Destinations.

Here is an example of some POLDO (Plain Old Listing Destination Output) I created for a lecture I gave last fall to students in a graduate research methods seminar at my alma mater, The George Washington University in Washington, DC. The rows in this report show statistics generated from four separate Analysis of Variable models created using the ANOVA Procedure in the SAS/STAT module. Using many of the Output Destination tools discussed above, I created a small SAS Data Set containing just the pieces of the PROC ANOVA output I wanted to discuss with the students. Here's that Data Set, “delivered” to the default ODS Listing Destination using PROC PRINT.



Obs	Name of Dependent Variable	Name of Independent Variable	Degrees of Freedom	F-Statistic Value	Probability Value	R Square	Dependent Variable Mean
1	Satis. w/Job	DETACHMENT	9	2.05	0.0428	0.176875	3.650000
2	Satis. w/Salary & Benefits	DETACHMENT	9	0.92	0.5131	0.087709	3.656250
3	Satis. w/Command. Offer.	DETACHMENT	9	7.42	<.0001	0.437142	3.979167
4	Satis. w/Co-Workers	DETACHMENT	9	0.99	0.4541	0.093904	4.197917

Figure 9: Plain Old Listing Destination Output (POLDO)

While the results (i.e., the “numbers”) are just what I want, I'd prefer to share these results as in Portable Document File output and “pretty-up” the presentation using an ODS Style Template. Using the ODS and PROC PRINT statements below, I was able to have ODS “deliver” the output to a PDF file that is styled with the SASWEB Style Template, one of the 43 internal, or “SAS-supplied” style templates available in SAS 9.1.3 software. Notice that the LISTING destination is closed before opening the PDF destination. After the PROC PRINT step, the PDF destination

is closed and the LISTING destination is re-opened. Also, the STYLE option is used in the ODS PDF Statement to select the Style Template I want to use. If you don't specify a Style Template with one of the "Presentation Destinations," SAS will automatically use the Default Style.

```

218 ods listing close;
219 ods pdf file = 'c:\combine1.pdf' style=sasweb;
220 proc print data=combine label;
221 format dependent $depvarf.;
222 title5 'Combining ODS Objects in to One SAS Data Set';
223 run;
224 ods pdf close;
225 ods listing;

```

Figure 10: Using the PDF Destination with the SASWEB Style Template

The output looks like this:

GWU School of Business
Decision Science 276: Exploratory and Multivariate Data Analysis
Lecture 4: Fall 2007 Semester
Introduction to Multivariate Analysis
Combining ODS Objects in to One SAS Data Set

Obs	Name of Dependent Variable	Name of Independent Variable	Degrees of Freedom	F-Statistic Value	Probability Value	R Square	Dependent Variable Mean
1	Satis. w/Job	DETACHMENT	9	2.05	0.0428	0.176875	3.650000
2	Satis. w/Salary & Benefits	DETACHMENT	9	0.92	0.5131	0.087709	3.656250
3	Satis. w/Command. Offcr.	DETACHMENT	9	7.42	<.0001	0.437142	3.979167
4	Satis. w/Co-Workers	DETACHMENT	9	0.99	0.4541	0.093904	4.197917

Figure 11: PROC PRINT Output "Delivered" to the PDF Destination and "Styled" with the SASWEB Style Template

STEP 6: KNOW YOUR STYLE

You can easily learn the names of available Style Templates by executing a brief PROC TEMPLATE step like the one shown below. The result is a list, in your Output Window, of the Style Templates you can use. The syntax is:

```

proc template;
list styles;
run;

```

If you are using SAS 9.1.3 Software, the results of this PROC TEMPLATE step are:

Obs	Path	Type
1	Styles	Dir
2	Styles.Analysis	Style
3	Styles.Astronomy	Style
4	Styles.Banker	Style
5	Styles.BarrettsBlue	Style
6	Styles.Beige	Style
7	Styles.Brick	Style
8	Styles.Brown	Style
9	Styles.Curve	Style
10	Styles.D3d	Style
11	Styles.Default	Style
12	Styles.EGDefault	Style
13	Styles.Education	Style
14	Styles.Electronics	Style
15	Styles.Festival	Style
16	Styles.FestivalPrinter	Style
17	Styles.Gears	Style
18	Styles.Journal	Style
19	Styles.Magnify	Style
20	Styles.Meadow	Style
21	Styles.MeadowPrinter	Style
22	Styles.Minimal	Style
23	Styles.Money	Style
24	Styles.NoFontDefault	Style
25	Styles.Normal	Style
26	Styles.NormalPrinter	Style
27	Styles.Printer	Style
28	Styles.Rsvp	Style
29	Styles.Rtf	Style
30	Styles.Sasweb	Style
31	Styles.Science	Style
32	Styles.Seaside	Style
33	Styles.SeasidePrinter	Link
34	Styles.Sketch	Style
35	Styles.Statdoc	Style
36	Styles.Statistical	Style
37	Styles.Theme	Style
38	Styles.Torn	Style
39	Styles.Watercolor	Style
40	Styles.blockPrint	Style
41	Styles.fancyPrinter	Style
42	Styles.sansPrinter	Style
43	Styles.sasdocPrinter	Style
44	Styles.serifPrinter	Style

Figure 12: List of ODS Style Templates Available in SAS 9.1.3 Software Generated by Using the LIST STYLES Statement in PROC TEMPLATE.

My friend Tasha Hodges of the State of Oregon's Division of Consumer and Business Services took the time recently to create a PDF that shows how a page of PROC PRINT output is "styled" using each of these Templates. You can download a copy of her PDF from my company web site from the "User Group Presentations" link at www.Sierrainformation.com. Using her examples, as well as a little trial and error on your part, can help you pick the Template that is best for your particular project or program.

It's possible to create customized Style Templates if one of these 43 "built in" Styles aren't quite what you need. In most cases, you can take a SAS-supplied Template and modify it a bit to create your Style, which is likely to save you a lot of coding to build a Style "from the ground up." For more information on how to create a customized Style Template please read Lauren Hawiorth's paper "ODS with Style: Creating Your Own ODS Style Template," a link to which is provided in the References section below.

STEP 7: KNOW YOUR STYLE STATEMENTS

Customizing your "presentation destination" reports does not always require taking the time and effort to create your own ODS Style Template. If you can use PROC REPORT, TABULATE or PRINT to display your output, mastering the use of ODS Style Statement in those BASE SAS Procedures can help you customize your reports quickly and easily. More details and insights in to the use of ODS Style Statements are available in the documentation manual

chapters for these Procedures and by reading the SUGI/SAS Global Forum papers by Lauren Haworth and Pete Lund listed in the References section below.

Here is an example of using an ODS Style Statement to customize the appearance of the PROC PRINT Output shown in Figure 11. Although we “like” how the SASWEB Style displays the output, we want the background color in the column-header to be black, rather than blue, since black prints out a bit better on the office laser printer. Instead of going to the time and effort to write a Style Template to make this modification, all we need to do is add the appropriate ODS Style Statements in a PROC REPORT Step. While I could have done the same thing in PROC PRINT, my next example will show another ODS STYLE Statement functionality that builds on this example to include conditionally assigning colors to the data in one of the report’s columns.

The code is:



```

227 options orientation=landscape;
228 ods pdf file = 'c:\pdf1.pdf' style=sasweb;
229 ods listing close;
230 proc report nowindows
231     data=combine
232     style(header)=[font_weight=bold background=black foreground=white]
233     style=sasweb split = *;
234 column dependent source df depmean rsquare fvalue probf;
235 define dependent/display format=$depvarf. 'Dependent*Variable*Name' width=45;
236 define source/display 'Independent*Variable*Name' width=12;
237 define df / display width = 13 'Degrees*of*Freedom' width = 10;
238 define depmean / display format=8.2 'Mean of*Dependent*Variable' width = 12;
239 define rsquare / display 'R-Square' width = 8 format=8.4;
240 define fvalue / display 'ANOVA*F*Statistic' width = 10;
241 define probf / display 'Prob. of*Type I*Error' width = 10;
242 title5 'Using PROC REPORT';
243 run;
244 ods pdf close;
245 ods listing;

```

Figure 13: Using an ODS STYLE Statement in PROC REPORT

Notice the syntax on line 232 of the Program Editor. This example of an ODS Style Statement instructs ODS to change the background color of the column-headers to black and the foreground color to white. And, please take a look at line 228. Here we are assigning SASWEB as the template to “style” the entire report. By coding the ODS STYLE Statement on line 232, ODS will “honor” all the style commands in the SASWEB Style except for those controlling the appearance of the report’s column-headers, which will be “styled” in accordance with the commands in the Style Statement. When the boss springs for a fancy new color laser printer, we can easily change the syntax on line 232 to re-set the background color of the column headers to something other than black without having to re-write a customized Style Template.

STEP 8: USE ODS FOR TRAFFIC-LIGHTING

Here is a final introductory Step to Success with the SAS Output Delivery System. Combining a series of ODS and other SAS Software capabilities makes it very easy to generate reports that “traffic-light,” or conditionally assign appearance attributes to the report as a function of the data values in the report. For more details on these ODS features, please see two papers available for download from the User Group Presentations section at www.sierrainformation.com: “Getting Un-Stuck in the Traffic-Lighting” and “Traffic-Lighting Your Reports the Easy Way with PROC REPORT and ODS.” Here is a brief example of how a SAS Value Format is first used to assign colors to ranges of values. Then, that value format is supplied in an ODS Style Statement in a PROC REPORT step to have ODS assign colors to the values appearing in that row of the report. First, here is the PROC FORMAT step that creates the Value Format implementing our “decision rules” about which ranges of values are assigned which colors:

```

198 proc format;
199 * assign format labels to dependent variable names;
200 * note: dollar sign ($) precedes name of format associated to;
201 * values of character variables;
202 value $depvarf SATF1 = 'Satis. w/Job'
203             SATF2 = 'Satis. w/Salary & Benefits'
204             SATF3 = 'Satis. w/Command. Offcr.'
205             SATF4 = 'Satis. w/Co-Workers';
206 * assign colors to p-values;
207 value pvalf low - .0299999999 = 'green'
208             .03 -< .0999999999 = 'blue'
209             .10 - high = 'red';
210 run;

```

Figure 14: Assigning Colors to Ranges of Values

Now that we have the Value Format we need, the next step is to add the appropriate ODS Style Statement in to our PROC REPORT task that that the commands in the Format are used to assign colors to a column in the output report. The code is:

```

248 options orientation=landscape;
249 ods pdf file = 'c:\pdf2.pdf' style=sasweb;
250 ods listing close;
251 proc report nowindows
252     data=combine
253     style(header)=[font_weight=bold background=black foreground=white]
254     style=sasweb split = '*';
255 column dependent source df depmean rsquare fvalue probf;
256 define dependent/display format=$depvarf. 'Dependent*Variable*Name' width=45;
257 define source/display 'Independent*Variable*Name' width=12;
258 define df / display width = 13 'Degrees*of*Freedom' width =10;
259 define depmean / display format=8.2 'Mean of*Dependent*Variable' width = 12;
260 define rsquare / display 'R-Square' width = 8 format=8.4;
261 define fvalue / display 'ANOVA*F*Statistic' width = 10;
262 define probf / display 'Prob. of*Type I*Error' width = 10
263     style(column)=[font_weight=bold foreground=pvalf.];
264 title5 'Using PROC REPORT';
265 run;
266 ods pdf close;
267 ods listing;

```

Figure 15: Using an ODS Style Statement to “Traffic-Light” a Column in PROC REPORT

Looking at line 263 of the Program Editor extract above, we can see that the PVALF Format is specified to control the foreground colors in one column in the PROC REPORT-generated output. As the PROC builds the report, the values in the PROBF column are “traffic-lit” according to the commands in the PVALF Format. The output looks like this:

GWU School of Business
 Decision Science 276: Exploratory and Multivariate Data Analysis
 Lecture 4: Fall 2007 Semester
 Introduction to Multivariate Analysis
 Using PROC REPORT



Dependent Variable Name	Independent Variable Name	Degrees of Freedom	Mean of Dependent Variable	R-Square	ANOVA F Statistic	Prob. of Type I Error
Satis. w/Job	DETACHMENT	9	3.65	0.1769	2.05	0.0428
Satis. w/Salary & Benefits	DETACHMENT	9	3.66	0.0877	0.92	0.5131
Satis. w/Command. Offcr.	DETACHMENT	9	3.98	0.4371	7.42	<.0001
Satis. w/Co-Workers	DETACHMENT	9	4.20	0.0939	0.99	0.4541

Figure 16: Traffic-Lighting Results

CONCLUSION

In this paper I have tried to highlight what I feel are some of the core functionalities of the SAS Output Delivery System and how they can be used to both simplify and enhance your projects and programs. ODS is a vast and growing element of the SAS System, and significant enhancements to it are included in the forthcoming release of SAS 9.2 Software. If you start with the basics, and master the core concepts of how ODS works, you can easily apply what it offers to improve your reports and analyses while reducing the above of "labor" required to complete your SAS-generated reports.

REFERENCES

One of the best ways to stay current on ODS tools is to visit the ODS Section of the "BASE SAS Community" at <http://support.sas.com/rnd/base/ods/index.html>. This section page contains numerous links to information about current and forthcoming ODS features.

Here are some papers I've referenced in this paper by other authors. The links are to their papers in the Online Proceedings section at www.sasglobalforum.org

Haworth, Lauren, "SAS with Style: Creating Your Own ODS Style Template," <http://www2.sas.com/proceedings/sugi30/132-30.pdf>

Haworth, Lauren, "ODS for PRINT, REPORT and Tabulate," <http://www2.sas.com/proceedings/sugi26/p003-26.pdf>

Lund, Pete, "PDF Can Be Pretty Darn Fancy," <http://www2.sas.com/proceedings/sugi31/092-31.pdf>

A number of ODS-related papers (as PDFs) are available for free download from my company web site at http://www.sierrainformation.com/html/past_presentations_list.php. They include

- ODS 101: Essential Concepts of the SAS Output Delivery System
- ODS 201: Making Your Life Easier with the SAS Output Delivery System
- Getting Un-Stuck in the Traffic-Lighting
- Traffic-Lighting Your Reports the Easy Way with PROC REPORT and ODS
- Examples: SAS-Supplied ODS Style Templates in 9.1 (courtesy of Tasha Hodges, Oregon Department of Consumer and Business Services)

ACKNOWLEDGEMENTS

I would like to thank Jennifer Yammamoto of the Cancer Research Center of Hawai'i for sharing the lung cancer patient data set which I used for several of the examples in this paper. Philip W. Wirtz, Ph.D., Professor of Decision

Sciences and of Psychology at The George Washington University, from whom I first learned SAS more years ago than either of us want to admit, was kind enough to have me present an early version of this paper to graduate students in his Fall 2007 quantitative methods seminar. Tasha Hodges of the State of Oregon DCBS took the time to generate examples of output “styled” with all 43 SAS 9.1 Style Templates and to then permit me to make her work available to other SAS users via my company web site.

I also want to acknowledge the willingness of many SAS Institute employees, including Sandy McNeill, David Kelley, Chevell Parker and Bob Rodriguez, from whom I have learned a great deal about the internal workings of ODS to both share their expertise on these topics with me and to attend many of my user group presentations on ODS-related topics.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Andrew H. Karp
Sierra Information Services
19229 Sonoma Hwy PMB 264
Sonoma, CA 95476
Email: Andrew@sierrainformation.com
Web: www.SierraInformation.com

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.
Other brand and product names are trademarks of their respective companies.