ODS, An Introduction to Creating Output Data Sets
Lisa Pyle, Merck & Co., Inc., North Wales, PA

ABSTRACT
Not using ODS yet in your SAS® programs? This is a brief introduction to help you identify procedure output table names, find your procedure output data and manipulate the data for reporting. Examples will help beginners quickly adapt to using ODS generated output data sets from procedures in their programs.

INTRODUCTION
In the pharmaceutical arena or any industry where programmers need to access procedure information for further manipulation and reporting, it is important to be able to access results quickly and accurately. When performing data manipulation and statistical analyses using procedures, you can use traditional ways of outputting data to data sets or you can use the easier approach of the Output Delivery System (ODS).

Introduced in production in Version 8, the Output Delivery System is used by all SAS procedures to produce ODS output objects or binary objects that can be rendered to various output destinations. An output object consists of two component parts: a data object containing the raw data values for the piece of output, and a template describing how the piece of output should look. Using ODS allows you to quickly obtain information in data sets to be further processed and provides for more flexibility rather than traditional methods of procedure data access.

CREATING OUTPUT DATA SETS
Creating SAS data sets from an output object is fairly straight-forward. You merely precede your procedure with the following SAS code:

```
ODS OUTPUT output-object-specification<=SAS-data-set>;
```

The output-object-specification identifies the output object to turn into a SAS data set. The output-object-specification can be specified as a full path, a partial path, a label, a label path, a mixture of labels and paths or the name. For me, the easiest way to specify the output-object is by using the name. More often than not, the name is a partial path so there's no rocket science involved. Plus, it's the first item listed on the trace record so it's easy to identify.

IDENTIFYING THE OUTPUT OBJECT
How do you determine what output object or objects a procedure creates and what is the trace record? Again, this is pretty easy. The SAS statement:

```
ODS TRACE ON;
```

when preceded by a SAS procedure quickly provides a method for identifying the output object(s) created. The example code below will produce a trace record for the PROC MEANS statement or will write to the SAS log a record of each output object that is created for this procedure.

```
ods trace on;
proc means data=bp;
var bpsys bpdia;
run;
ods trace off;
```

The information or the Trace Record is output to the Log window and is depicted in Figure 1.
Notice in Figure 1 that the PROC MEANS example only produces one output object. In this case, it is clear that the summary information provided by the PROC MEANS statement will all be contained in this one output object named ‘Summary’. However, this may not always be the case for all procedures.

Having the trace record output to the Log window is a useful start, but it can quickly be cumbersome if you have a procedure with lots of output objects either due to the complexity of the procedure itself or the complexity of the options requested by you.

For some more straightforward procedures like PROC MEANS, there may only be one output object. In which case, using the above ODS TRACE ON statement is fine because you can easily identify to which output destination the procedure information is stored. On the other hand, for more complicated procedures the output may be split into more than one output destination. That is, in some cases a procedure will produce several output objects because not all of the output can be stored into one output object. Instead, the output may be divided into different output objects depending on the topic of the output or structure.

Let's take a look at a regression analysis. Following is the code for an example regression analysis. The regression analysis results that go to the Output window are shown in Figure 2 and the corresponding Trace Record that is output to the LOG window is shown in Figure 3.

```sas
ods trace on;
proc reg data=bp ;
model map=weight gender;
run;
ods trace off;
```

For this PROC REG procedure, the three output objects are produced: ‘ANOVA’, ‘FitStatistics’ and ‘ParameterEstimates’ as seen in Figure 3. When there is more that one output object produced by a procedure, it may be difficult to pin-point which output object contains the procedure information that you may need if you do not have a statistical background.

In fact, if you merely look at the regression output in Figure 2, the output is only labelled or subdivided into two sections: ‘Analysis of Variance’ and ‘Parameter Estimates’. The ‘Fit Statistics’ are not labelled, so without having the trace record displayed, you may not immediately know into what output object this data will fall. So as analyses get more complicated, it is useful to have another tip on hand.
FIGURE 2

The REG Procedure
Model: MODEL1
Dependent Variable: MAP Mean Arterial Pressure

Analysis of Variance

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>43.47753</td>
<td>21.73876</td>
<td>0.25</td>
<td>0.7773</td>
</tr>
<tr>
<td>Error</td>
<td>76</td>
<td>6536.69125</td>
<td>86.00910</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>78</td>
<td>6580.16878</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Root MSE = 9.27411, R-Square = 0.0066, Dependent Mean = 106.94093, Coeff Var = 8.67218

Parameter Estimates

| Variable | Label   | DF | Parameter Estimate | Standard Error | t Value | Pr > |t| |
|----------|---------|----|--------------------|----------------|---------|------|---|
| Intercept| Intercept| 1  | 108.07091          | 5.98848        | 18.05   | <.0001|
| weight   |         | 1  | -0.01125           | 0.04057        | -0.28   | 0.7823|
| gender   |         | 1  | 1.50073            | 2.21071        | 0.68    | 0.4993|

FIGURE 3

Output Added:
-------------
Name: ANOVA
Label: Analysis of Variance
Template: Stat.REG.ANOVA
Path: Reg.MODEL1.Fit.MAP.ANOVA
-------------

Output Added:
-------------
Name: FitStatistics
Label: Fit Statistics
Template: Stat.REG.FitStatistics
Path: Reg.MODEL1.Fit.MAP.FitStatistics
-------------

Output Added:
-------------
Name: ParameterEstimates
Label: Parameter Estimates
Template: Stat.REG.ParameterEstimates
Path: Reg.MODEL1.Fit.MAP.ParameterEstimates
-------------
In this situation and others where it is not necessarily clear into what output object data resides, an easier way to identify where procedure output data is located is to add the following option to the ODS Trace statement:

```
ODS TRACE ON /listing;
```

In Figure 4 with the addition of the /listing option, the trace record is written to the Output window instead of the Log window. Furthermore, each part of the trace record immediately preceds the output object that is describes so there is no confusion or guessing as to what data resides in what output object. It's now crystal clear!

---

### FIGURE 4

The REG Procedure
Model: MODEL1
Dependent Variable: MAP Mean Arterial Pressure

Output Added:
-------------
Name: ANOVA
Label: Analysis of Variance
Template: Stat.REG.ANOVA
Path: Reg.MODEL1.Fit.MAP.ANOVA
-------------

Analysis of Variance

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2</td>
<td>43.47753</td>
<td>21.73876</td>
<td>0.25</td>
<td>0.7773</td>
</tr>
<tr>
<td>Error</td>
<td>76</td>
<td>6536.69125</td>
<td>86.00910</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>78</td>
<td>6580.16878</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output Added:
-------------
Name: FitStatistics
Label: Fit Statistics
Template: Stat.REG.FitStatistics
Path: Reg.MODEL1.Fit.MAP.FitStatistics
-------------

Root MSE 9.27411  R-Square 0.0066
Dependent Mean 106.94093  Adj R-Sq -0.0195
Coeff Var 8.67218

Output Added:
-------------
Name: ParameterEstimates
Label: Parameter Estimates
Template: Stat.REG.ParameterEstimates
Path: Reg.MODEL1.Fit.MAP.ParameterEstimates
-------------

Parameter Estimates

| Variable | Label | DF | Parameter Estimate | Standard Error | t Value | Pr > |t| |
|----------|-------|----|--------------------|----------------|---------|------|---|
| Intercept| Intercept | 1  | 108.07091          | 5.98848        | 18.05   | <.0001 |
| weight   |       | 1  | -0.01125           | 0.04057        | -0.28   | 0.7823|
| gender   |       | 1  |                    |                |         |      |
By looking at the output in Figure 4, you clearly know that the Root MSE, the Dependent Mean and the Coeff Var are located in the output object FitStatistics. There is no guessing and you can immediately determine whether or not you will need to use this output object or not in your subsequent programming.

CREATING OUTPUT DATA SETS REVISITED

Now that you know how to identify the output objects using tracing, you can go ahead and create data sets. Recall that there are a number of ways to specify the procedure information. Referring to the earlier PROC MEANS example, the easiest is to use the output object name or 'Summary'. In this sample code, you provide summary statistics for two variables bpsys and bpdia in the data set bpmeas.

```sas
ods output summary=bpmeas;
proc means data=bp;
var bpsys bpdia;
run;
```

The data sets can be manipulated like any other SAS data set. You can include or exclude components of the analysis by subsetting the data sets using data set options in parentheses right after the SAS data set. For example, even though you provide summary statistics for both variables bpdia and bpsys, you may decide that you only need to report the MEAN and Standard Deviation for one of the variables. In this case, you could use data set options to filter for the information for only one variable.

```sas
ods output summary=bpmeas(keep=bpsys_mean bpsys_stddev);
proc means data=bp;
var bpsys bpdia;
run;
```

Unlike the PROC MEANS procedure, the PROC UNIVARIATE procedure structures the output objects vertically when summary statistics on multiple variables are requested. When this occurs, you can use the output object path name to specify the information related to one variable. For example, to obtain the basic summary statistics for the bpsys variable you can specify the path and then the output data set mom2 will contain the results for the variable bpsys only.

```sas
ods output Univariate.BPSYS.BasicMeasures=mom2;
proc univariate data=bp;
var bpsys bpdia;
run;
```

Not only can you work with multiple variables but you can also work with multiple models. For example, you may perform several regression analyses, but you may ultimately only report the Parameter Estimates for only one of the models. In this example, the parameter estimates for both models are available in the data set peinfo for further evaluation.

```sas
ods output parameterestimates=peinfo;
proc reg data=bp;
model map=weight gender;
model map=weight;
quit;
```

To subset for only the information from the first model, merely add a data set option to filter the resulting data set. The analysis for both models will still appear in the output window for reference, but the data set will only contain the information that you will need for reporting or manipulation.

```sas
ods output parameterestimates=peinfo(where=(model='MODEL1'));
proc reg data=bp;
model map=weight gender;
model map=weight;
```
The ODS OUTPUT statement is even flexible to allow for more than one output object to be specified. So for example, if you are interested in manipulating information from the Fit Statistics and the Parameter Estimates you can specify to output the results into data sets for both of these output objects at the same time.

```plaintext
ods output parameterestimates=peinfo fitstatistics=fs;
proc reg data=bp ;
    model map=weight gender;
quit;
```

LIMITING OUTPUT

Now that you are familiar with how to identify the output objects that result from a given procedure, you can manipulate the output even more. Beyond creating data sets by using ODS output, you can also subset the output that is visible in the Output window. By using the ODS SELECT and ODS EXCLUDE statements you can limit the results printed to the Output window.

Let's revisit the regression analysis in Figure 2. If after reviewing results and considering what is needed for reporting, you decide that all that you are interested in retaining is the analysis of variance output from the PROC REG, you can limit the output using the ODS SELECT statement.

```plaintext
ods select anova;
proc reg data=bp ;
    model map=weight gender;
quit;
```

By preceding the regression analysis code with the ODS SELECT statement, the Output window will only print the results of the ANOVA and will exclude all the other output. That is the fit statistics and the parameter estimates will not be printed to the Output window. On a similar note, ODS EXCLUDE will omit specific procedure output from the Output window.

```plaintext
ods exclude anova;
proc reg data=bp ;
    model map=weight gender;
quit;
```

In the code above, the results from the Fit Statistics and the Parameter Estimates will be visible in the Output window and the ANOVA results will be omitted. This flexibility of limiting information to the Output Window to only information you are interested in can greatly simplify your review of procedure results.

CONCLUSION

Selecting and manipulating data from procedures is greatly simplified with the use of ODS output objects. With a few simple steps and some additional code, you have the power to take and manipulate elements of procedure output. Using tracing can allow you to quickly identify in what output object or objects procedure results reside so that you can select the information that you need. Combining this with ODS output provides you with the flexibility to store the procedure information into data sets so that you have the flexibility to perform whatever additional functions you require. Further, with ODS SELECT and EXCLUDE you have the flexibility to focus in on important components of the analysis results. These programming tips allows you to quickly filter thru procedure output and accurately access data for further analyses and reporting.

REFERENCES


SAS Institute Inc., SAS OnlineDoc Version 8.1

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

Lisa Pyle
Merck & Co., Inc.
P.O. Box 1000
North Wales, PA 19454-1099
Work Phone: 267-688-0167
Fax: 267-305-6474
E-mail: lisa_pyle@merck.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.