

## There's Nothing ODious about ODS

Aaron J. Rabushka, inVentiv Health Clinical, Austin, TX USA

### ABSTRACT

This presentation provides a look at SAS® ODS capabilities with regard to enhancing output, suppressing output, and drawing information from SAS PROCs into working data sets. Its code examples were tested under SAS V9.1.3, SP4.

### INTRODUCTION

What is ODS? Some things that it is not:

- it is not a reference to Operation Desert Storm
- it is not some kind of Operational Data set Storage
- it is not an On-line Dating Service
- it is not "Onward Data set Soldiers"

What it does stand for is "Output Delivery System," which is a system that handles all SAS output whether or not the user issues any specific commands for it. SAS introduced user control over ODS in Version 7, and the examples in this presentation work under Version 9.1.3, SP 4. Note that color highlights in the examples come from the author of this presentation and do not necessarily reflect what will come up logs of other output from actual SAS runs.

### USING ODS TO SUPPRESS PRINTED OUTPUT

What do you do when you want the output from a SAS PROCedure in a data set but not in any printed form? It is well known that some SAS PROCs (e.g., UNIVARIATE and FREQ) offer the NOPRINT option, whereas others (e.g., TABULATE) do not.

To suppress printed output, frame the PROC code with the following ODS LISTING statements:

```
ODS LISTING CLOSE;

PROC TABULATE DATA=EXAMPLE_DATA OUT=DEMOSTATS (DROP=_TYPE_ _PAGE_ _TABLE_);
CLASS RACE SEX;
VAR AGE;
TABLES (SEX ALL), RACE*AGE*(N MEAN STD MAX*F=2. MIN*F=2.);
RUN;

ODS LISTING;
```

### USING ODS TO MODIFY PRINTED OUTPUT

ODS can help rid your outputs of the "it looks like it came off a computer" feel. You can use it to direct outputs to several types of *destinations*, including HTML, RTF, XML, and (Adobe) PDF.

To direct output to HTML, frame your PROC code with the following ODS statements:

```
ODS HTML STYLE=MINIMAL FILE="C:\AJROUTPUT\DEMOFREQ.HTML";

PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;
  TABLES AGE;
  FORMAT AGE AGEGRP.;
  TITLE "HTML OUTPUT FROM SAS ODS";
```

## *There's Nothing ODiouS about ODS, continued*

```
RUN;
```

```
ODS HTML CLOSE;
```

To direct it to PDF:

```
ODS PDF STYLE=MINIMAL FILE="C:\AJROUTPUT\DEMOFREQ.PDF";
```

```
PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;  
  TABLES AGE;  
  FORMAT AGE AGEGRP.;  
  TITLE "PDF OUTPUT FROM SAS ODS";  
RUN;
```

```
ODS PDF CLOSE;
```

...and to RTF:

```
ODS RTF BODY='TEMP.RTF' STYLE=STYLES.TEST FILE="C:\AJROUTPUT\DEMOFREQ.RTF";
```

```
PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;  
  TABLES AGE;  
  FORMAT AGE AGEGRP.;  
  TITLE '\R\I THESE ARE THE \LINE EXAMPLE DATA\LINE IN RTF';  
RUN;
```

```
ODS RTF CLOSE;
```

Each of these presents the output in a slightly different style.

You can use ODS intricately with PROC REPORT to enhance the printed results. Here is some code that will format output on a cell-by-cell basis:

```
ODS RTF BODY='TEMP.RTF' STYLE=STYLES.TEST FILE="C:\AJROUTPUT\DEMOREPORT1.RTF";  
PROC REPORT DATA=DATAOUT.EXAMPLE_DATA MISSING HEADLINE HEADSKIP SPLIT='~'  
  SPACING=1 NOWD ;  
  WHERE AGE > 65;  
  COLUMNS ("DEMOGRAPHIC VARIABLES" SEX AGE RACE) DOSE;  
  DEFINE SEX / ORDER WIDTH=20 ORDER=INTERNAL STYLE(COLUMN) = [FONT_STYLE = ITALIC];  
  DEFINE AGE / ORDER WIDTH= 20 ORDER=INTERNAL;  
  DEFINE RACE / WIDTH = 20 DISPLAY  
    STYLE(COLUMN) = [JUST=CENTER FOREGROUND=RED FONT_STYLE=ITALIC];  
  DEFINE DOSE / DISPLAY STYLE(COLUMN)=[FONT_WEIGHT=BOLD];  
  TITLE1 "PROC REPORT OUTPUT";  
  TITLE2 "INTO RTF";  
RUN;  
ODS RTF CLOSE;
```

Here are the results:

**PROC REPORT OUTPUT  
INTO RTF**

DEMOGRAPHIC VARIABLES			
Sex (Text)	Age (yr)	Race (Text)	dose
<i>FEMALE</i>	68	<i>WHITE</i>	<b>3</b>
	69	<i>WHITE</i>	<b>1</b>
	71	<i>WHITE</i>	<b>2</b>
		<i>WHITE</i>	<b>1</b>
	73	<i>WHITE</i>	<b>8</b>
	75	<i>WHITE</i>	<b>6</b>
	76	<i>OTHER: SPANISH</i>	<b>4</b>
	79	<i>WHITE</i>	<b>4</b>
	82	<i>WHITE</i>	<b>3</b>
<i>MALE</i>	66	<i>WHITE</i>	<b>8</b>
	67	<i>WHITE</i>	<b>4</b>
	70	<i>BLACK</i>	<b>3</b>
	71	<i>WHITE</i>	<b>3</b>
	75	<i>ASIAN</i>	<b>6</b>
	77	<i>WHITE</i>	<b>3</b>
		<i>WHITE</i>	<b>7</b>
	78	<i>WHITE</i>	<b>2</b>

**Figure 1: PROC REPORT output enhanced through ODS**

Some use proprietary macros to further enhance the interaction between ODS and other programs outside of SAS (e.g., Word, Excel).

**USING ODS TO CREATE SAS DATA SETS FROM SAS PROCs**

You can use ODS to pull information from SAS PROCedures into SAS data sets even if it does not show up in the procedure's OUTPUT data sets. You use **ODS TRACE** to find out which objects are available from a SAS PROC. For example:

## There's Nothing ODiouS about ODS, continued

```
ODS TRACE ON;
PROC CORR DATA=DATAOUT.EXAMPLE_DATA;
  VAR AGE DOSE;
  TITLE1 "PROC CORR OUTPUT";
RUN;
ODS TRACE OFF;
```

This gives the following results in the log, which shows the attributes of ODS objects:

```
298 ODS TRACE ON;
299
300 PROC CORR DATA=DATAOUT.EXAMPLE_DATA;
301   VAR AGE DOSE;
302   TITLE1 "PROC CORR OUTPUT";
303 RUN;
Output Added:
-----
Name:      VarInformation
Label:     Variables Information
Template:  base.corr.VarInfo
Path:      Corr.VarInformation
-----
Output Added:
-----
Name:      SimpleStats
Label:     Simple Statistics
Template:  base.corr.UniStat
Path:      Corr.SimpleStats
-----
Output Added:
-----
Name:      PearsonCorr
Label:     Pearson Correlations
Template:  base.corr.StackedMatrix
Path:      Corr.PearsonCorr
-----

NOTE: PROCEDURE CORR used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds
304
```

These attributes are the following:

**NAME:** name of the object,  
e.g., **PearsonCorr** .

**LABEL:** a label for the object--usually resembles the name and takes it a step closer to readable English,  
e.g., **Pearson Correlations**.

**TEMPLATE:** a template for display—this can be modified with PROC TEMPLATE,  
e.g., **base.corr.StackedMatrix** .

**PATH:** a path to the object which recaps the name as its rightmost element,  
e.g., **corr.PearsonCorr** .

```
305 ods trace off;
```

In order to use ODS OUTPUT to put ODS objects into SAS data sets, identify the object by its NAME, and, if necessary, by its PATH. You can place the ODS OUTPUT statement either before or inside the PROC for which you are extracting the object. Note that the assignment statements associated with ODS OUTPUT take their place among SAS's "backward" assignment statements where the name of the data set being assigned is placed to the right of the equal sign. (NOBS = in the SET statement operates in a similar manner). Both of the following give the same results:

## There's Nothing ODiouS about ODS, continued

```
ODS OUTPUT PEARSONCORR=CORRVAR;
PROC CORR DATA=DATAOUT.EXAMPLE_DATA OUTP=PEARSON_OUT;
  VAR AGE DOSE;
  TITLE1 'PROC CORR OUTPUT';
RUN;

AND

PROC CORR DATA=DATAOUT.EXAMPLE_DATA OUTP=PEARSON_OUT;
  VAR AGE DOSE;
  TITLE1 'PROC CORR OUTPUT';
  ODS OUTPUT PEARSONCORR=CORRVAR;
RUN;
```

Here is a log section that shows PROC CORR results output into SAS data sets, both through ODS and CORR's OUTF option:

```
307 ODS OUTPUT SIMPLESTATS=CORRSTATS PEARSONCORR=CORRVAR;
308
309
310 PROC CORR DATA=DATAOUT.EXAMPLE_DATA OUTP=PEARSON_OUT;
311   VAR AGE DOSE;
312   TITLE1 "PROC CORR OUTPUT";
313 RUN;
```

NOTE: The data set WORK.CORRVAR has 2 observations and 6 variables.  
NOTE: The data set WORK.CORRSTATS has 2 observations and 8 variables.  
NOTE: The data set WORK.PEARSON\_OUT has 5 observations and 4 variables.

Sometimes ODS can give you multiple objects with the same NAME, differentiated by PATH:

```
ODS TRACE ON;
PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;
  TABLES AGE RACE DOSE;
  FORMAT AGE AGEGRP.;
RUN;
```

- Output Added:
- -----
- Name: OneWayFreqs
- Label: One-Way Frequencies
- Template: Base.Freq.OneWayFreqs
- Path: Freq.Table1.OneWayFreqs
- -----
- Output Added:
- -----
- Name: OneWayFreqs
- Label: One-Way Frequencies
- Template: Base.Freq.OneWayFreqs
- Path: Freq.Table2.OneWayFreqs
- -----
- Output Added:
- -----
- Name: OneWayFreqs
- Label: One-Way Frequencies
- Template: Base.Freq.OneWayFreqs
- Path: Freq.Table3.OneWayFreqs
- -----

## There's Nothing ODiouS about ODS, continued

And the data sets are created as follows:

```
ODS OUTPUT ONEWAYFREQS = ALL_TABLES
      FREQ.TABLE1.ONEWAYFREQS = AGE_TABLE
      FREQ.TABLE2.ONEWAYFREQS = RACE_TABLE
      FREQ.TABLE3.ONEWAYFREQS = DOSE_TABLE
      ;
PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;
  TABLES AGE RACE DOSE;
  FORMAT AGE AGEGRP.;
RUN;
```

The output gives the following notices:

```
NOTE: The data set WORK.DOSE_TABLE has 8 observations and 7 variables.
NOTE: The data set WORK.RACE_TABLE has 4 observations and 7 variables.
NOTE: The data set WORK.AGE_TABLE has 6 observations and 7 variables.
NOTE: The data set WORK.ALL_TABLES has 18 observations and 11 variables.
NOTE: There were 57 observations read from the data set DATAOUT.EXAMPLE_DATA.
```

In PROC FREQ, ODS generates different objects from listed vs. crossed two-way tables:

```
ODS TRACE ON;
PROC FREQ DATA=DATAOUT.EXAMPLE_DATA;
  TABLES SEX*RACE;
  TABLES SEX*RACE / LIST;
RUN;
ODS TRACE OFF;
```

And the log shows:

```
Output Added:
-----
Name:          CrossTabFreqs
Label:         Cross-Tabular Freq Table
Data Name:
Path:          Freq.Table1.CrossTabFreqs
-----
```

```
Output Added:
-----
Name:          List
Label:         LIST Frequencies
Template:     Base.Freq.List
Path:         Freq.Table2.List
-----
```

ODS OUTPUT may not work with the NOPRINT option:

```
ODS OUTPUT PEARSONCORR=CORRVAR;
PROC CORR DATA=DATAOUT.EXAMPLE_DATA OUTP=PEARSON_OUT NOPRINT;
  VAR AGE DOSE;
RUN;
```

The log gives the following:

- 352 ODS OUTPUT PEARSONCORR=CORRVAR;
- 353
- 354
- 355 PROC CORR DATA=DATAOUT.EXAMPLE\_DATA OUTP=PEARSON\_OUT NOPRINT;
- 356 VAR AGE DOSE;
- 357 RUN;

## There's Nothing ODiouS about ODS, continued

- NOTE: The data set WORK.PEARSON\_OUT has 5 observations and 4 variables.
- NOTE: PROCEDURE CORR used (Total process time):
- real time 0.01 seconds
- cpu time 0.01 seconds
- WARNING: Output 'pearsoncorr' was not created. Make sure that the output object name, label, or path is spelled correctly. Also, verify that the appropriate procedure options are used to produce the requested output object. For example, verify that the NOPRINT option is not used.

To get ODS objects and suppress printed output you can write as follows:

```
ODS LISTING CLOSE;
ODS OUTPUT PEARSONCORR=CORRVAR;
PROC CORR DATA=DATAOUT.EXAMPLE_DATA OUTP=PEARSON_OUT;
    VAR AGE DOSE;
RUN;
ODS LISTING;
```

Note that ODS OUTPUT doesn't work with all PROCs. PROC PRINT is one with which it doesn't work.

- 379 ODS OUTPUT PRINT=AJRPRINT;
- 380
- 381 PROC PRINT DATA=DATAOUT.EXAMPLE\_DATA;
- 382 VAR AGE DOSE;
- 383 RUN;
- ERROR: PROC PRINT does not support the ODS OUTPUT statement.
- NOTE: The SAS System stopped processing this step because of errors.
- WARNING: Output 'print' was not created. Make sure that the output object name, label, or path is spelled correctly. Also, verify that the appropriate procedure options are used to produce the requested output object. For example, verify that the NOPRINT option is not used.
- NOTE: There were 57 observations read from the data set DATAOUT.EXAMPLE\_DATA.

Sometimes ODS can give you an object that runs but may not make sense. One example of this is the MOMENTS object from PROC UNIVARIATE, that puts unrelated statistics in the same observation.

## SUMMARY

In summary, ODS can be very useful to suppress printed output, enhance and modify printer output, and pull data out of printed output into SAS data sets.

## CONTACT INFORMATION

For a copy of the original slides (which have some more extensive coding and output examples) and all other questions, contact

Aaron J. Rabushka  
[aaron.rabushka@inventivhealth.com](mailto:aaron.rabushka@inventivhealth.com)  
phone: 512/666-5008

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 registered trademarks or trademarks of their respective companies.