

ODS For Dummies

Chris Olinger, SAS Institute Inc., Cary, NC

Abstract

The Output Delivery System (ODS) in Version 7 and Version 8 of the SAS® System provides a powerful feature set for customizing SAS output. This paper is primarily an introductory look at ODS for the person who is a beginner at SAS programming. Also included are some tips for more advanced SAS users based on feedback that we have received thus far.

Introduction

Are you a self-described Dummy? Do you look at the stack of manuals in the corner¹ and decide that you are hands-on learner? If so, this paper is for you. ODS can seem a bit overwhelming at first, but the fact is that it follows the 80-20 rule. You can learn the first 80% relatively painlessly. The remaining 20% requires a bit more effort. This paper is designed to show you how to do the most important things and to give you a glimpse at some of the more esoteric features.

The format of this paper is going to be FAQ oriented (Frequently Asked Questions) and will touch on the ODS items that users have been having the most questions about. In particular, this paper addresses several topics:

- what ODS is
- what an output object is
- creation of output, including flow of control
- How do I create?
 - HTML output
 - PDF output
 - RTF output
 - Output Data Sets
- Miscellaneous

ODS Overview

What is ODS? ODS is a system for delivering various forms of SAS output that are user configurable. The basic ODS component is called an *output object*. A procedure or data step creates one or more output objects during execution. These output objects are then rendered for the appropriate output destination as specified by the SAS user.

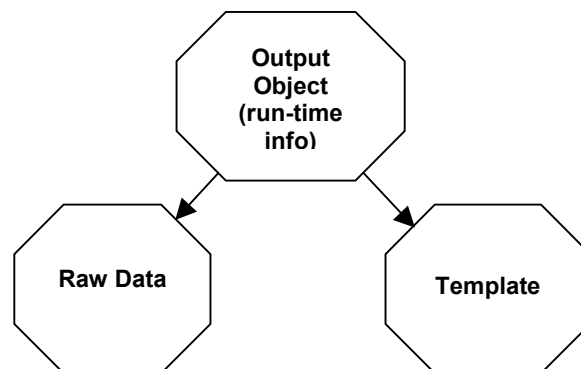
What is an output object? An output object is a basic representation of output, either tabular or graphical.

¹ "Shiny CD" doesn't have the same impact. I miss the manuals.

Most ODS output objects contain two basic components:

- a data component containing the raw data values that make up the piece of output
- and a template component

The output object also contains miscellaneous run-time information about the procedure that created the output. Represented visually, an output object looks like this:



What does the raw data represent? The raw data is a collection of numbers and strings that make up the table or graph. The data may be stored to an output data set or rendered as HTML, RTF, or some other output type.

What is a template? A template is a set of SAS statements and attributes that describes how ODS should format and present the raw data. There are several different types of templates in the system including tables, columns, and styles. You create and modify a template using PROC TEMPLATE. PROC TEMPLATE also provides statements to delete templates, list templates, and dump the SAS source statements that created a given template. You can find out more about the syntax for PROC TEMPLATE in *The Complete Guide to the SAS Output Delivery System*. You can also find out more information on templates by reading Olinger (1999).

Creation of Output

Output is produced by the procedures in the form of output objects. The output objects are then rendered to various output destinations based on the type of output requested by the SAS user.

Available types of output in Version 8 and Version 8.1 include:

- the standard SAS listing

- HTML
- PS, PDF, PCL
- RTF
- Output Data Sets
- XML

How Do I?

What follows are answers to some of the most common questions that we have received to date on ODS. It is not an exhaustive list, but it is enough to get you pointed in the right direction.

HTML:

How do I create HTML?

You create HTML via ODS by opening the ODS HTML destination:

```
ods html file="body.htm"
         contents="contents.htm"
         frame="frame.htm" ;

proc print data=sashelp.class; run;

ods html close;
```

The ODS statement is a global statement that can be specified anywhere in your SAS job. The above code creates three files: body.htm, contents.htm, and frame.htm. The body file contains the HTML that is created by the procedures that run between the open and close of the files. The contents file is a table of contents with links into the body file. And the frame file is a file that ties the body to the contents. Notice that there is nothing you need to do to your existing procedure code in order to enable HTML. Also, it is not required that you create all three files. If you want only the body file then leave off the contents and frame files.

The file specification may be a fileref (created with the FILENAME statement) as well as a quoted string. You can use a fileref to write HTML to just about anywhere.

How do I specify a different style for my HTML?

You can specify a different style for your HTML by using the STYLE option on the ODS HTML statement:

```
ods html file="body.htm"
         contents="contents.htm"
         frame="frame.htm"
         style=Styles.Beige;

... sas code ...

ods html close;
```

A style is collection of elements and attributes that designate the colors and fonts, among other things, that your HTML is going to use. Styles are stored along with templates in an ODS template store (a special type of SAS file). You can get a list of available styles by listing the styles in the current *template path*. The template path is where ODS looks to find templates and styles. The path is configured and queried using the ODS PATH statement. The TEMPLATE procedure allows you to list the various templates and styles:

```
1? ods path show;

Current ODS PATH list is:

1. SASUSER.TEMPLAT (UPDATE)
2. SASHELP.TMPLMST (READ)

2? proc template; list styles; run;

Listing of: SASHELP.TMPLMST
Path Filter is: Styles
Sort by: PATH/ASCENDING
Obs   Path                                     Type
-----
1     Styles                                     Dir
2     Styles.BarrettsBlue                       Style
3     Styles.Beige                               Style
4     Styles.Brick                               Style
5     Styles.Brown                               Style
6     Styles.D3d                                 Style
7     Styles.Default                             Style
8     Styles.Minimal                             Style
9     Styles.NoFontDefault                       Style
10    Styles.Printer                             Style
11    Styles.Rtf                                 Style
12    Styles.Statdoc                             Style
13    Styles.Theme                               Style
14    Styles.fancyPrinter                       Style
15    Styles.sansPrinter                         Style
16    Styles.sasdocPrinter                      Style
17    Styles.serifPrinter                       Style
```

As a shortcut on the ODS HTML statement you can remove the "styles." to specify the style. ODS will look first for a style named "Beige". If ODS can't find "Beige" then ODS will look for a style called "Styles.Beige". If ODS can't find either style then ODS will use Styles.Default:

```
ods html file="body.htm"
         contents="contents.htm"
         frame="frame.htm"
         style=Beige;

... sas code ...

ods html close;
```

Why is the generated HTML body file so large?

The ODS HTML driver was started at a time when browser companies were still at war with each other. HTML that worked with one browser did not work with other browsers. Also, certain browser's support of cascading style sheets (CSS) was also very poor. In order to generate HTML that worked everywhere without any problems, we decided to spit out enough information to make the HTML work correctly without using style sheets. Advances have been made more recently that allow greater use of style sheets. You can turn on this functionality directly on the ODS HTML statement:

```
ods html file="body.htm"
  contents="contents.htm"
  frame="frame.htm"
  style=Styles.Beige
  stylesheet="style.css" (url="style.css");

... sas code ...

ods html close;
```

The STYLESHEET option specifies the file to write the CSS code to. The optional URL sub-option specifies a URL to use to reference the file from the HTML body file. If you don't want to bother with a separate CSS file, you can embed the CSS file directly in your HTML file:

```
ods html file="body.htm"
  contents="contents.htm"
  frame="frame.htm"
  style=Styles.Beige
  css /* or "stylesheet" in 8.1 */;

... sas code ...

ods html close;
```

The CSS option will include the contents of the style sheet directly in the HTML body file. Use of style sheets will minimize the size of your HTML files. CSS support is experimental in release 8.0 and will be production in release 8.1.

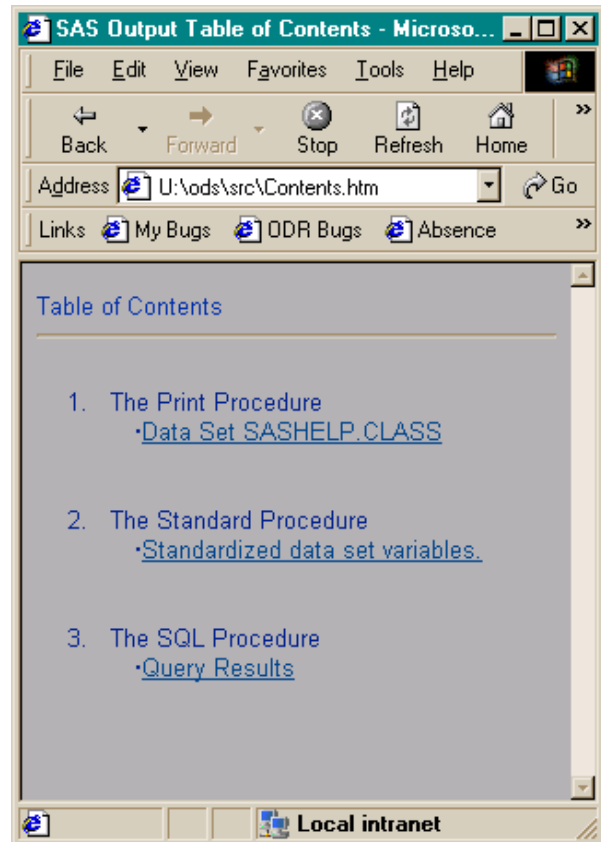
Also coming in release 8.1 is an experimental "bare-bones" version of HTML that can be gotten to via the ODS XML statement. Using the XML statement you'll be able to get small, concise HTML that relies on style sheets to format.

How do I change the table of contents?

This particular question has come up a lot.

Unfortunately, the answer is that the table of contents is not as configurable as some would like. We underestimated the desire of users to control every little detail. However, there are some things you can do to

effect some change. Below is a picture of a simple table of contents:



To change the appearance of the items in contents you need to create a new style or modify one that already exists. For more information on editing styles please see *The Complete Guide to the SAS Output Delivery System* or read the template paper by Olinger (1999).

To change the "Table of Contents" label, you must change the ContentTitle style element in the style:

```
proc template;
define style styles.ContentsChanges;
  parent=styles.default;
  style ContentTitle from ContentTitle /
  pretext = "Contents Listing";
end;
run;

ods html file="body.htm"
  contents="contents.htm"
  frame="frame.htm"
  style=Styles.ContentsChanges;

... sas code ...

ods html close;
```

The previous code will create a new style and override the definition for the ContentTitle style element.

To change the “The XXX Procedure” label you can take two different approaches. One is to modify the ContentProcName style element. The procedure name is fixed by the system. However, the prefix and suffix text is specified in the style element. The first part of the example below changes the label to “Listing For: XXX”. The second way to modify the procedure’s label is to use the ODS PROCLABEL statement. The ODS PROCLABEL statement specifies the label to use for the contents listing for the next procedure run:

```
/* first example */
proc template;
define style styles.ContentsChanges;
parent=styles.default;
style ContentProcName from ContentProcName /
pretext = "Listing For: "
posttext = ""
;
end;
run;

ods html file="body.htm"
contents="contents.htm"
frame="frame.htm"
style=Styles.ContentsChanges;

proc print data=sashelp.class; run;

/* second example */
ods proclabel "SQL is special";
proc sql; select * from sashelp.class; quit;
```

You can also change the label printed for each piece of output. Different procedures allow you to change this field in various ways. For the TABULATE procedure, the PRINT procedure, and the FREQ procedure cross table, you can set the label via the CONTENTS² option in the procedure’s syntax. For a SAS/GRAPH[®] procedure you can use the DESCRIPTION option. For all other output that uses an ODS template you can set the CONTENTS_LABEL³ option:

```
/* Contents option */
proc print data=sashelp.class
contents="CLASS Data Set";
run;

/* Description option */
proc gchart data=sashelp.class;
vbar3d height / description="Student Height";
run;
```

² You may also specify CONTENTS="" to remove the label from the table of contents.

³ Template is different. What can I say? The CONTENTS= option for template is an ON/OFF option to turn the label on and off. The CONTENTS_LABEL= option controls the text.

```
/* ProcTemplate */
proc template;
edit base.sql;
mvar cont;
contents_label=cont;
end;
run;

%let cont=List of SASHELP.CLASS;
proc sql; select * from sashelp.class;
quit;

ods html close;
```

This code and the previous code produce the following table of contents:

The screenshot shows a web browser window titled "SAS Output Table of Contents - Microsoft I...". The address bar contains "U:\ods\src\contents.htm". Below the address bar are navigation buttons: Back, Forward, Stop, Refresh, Home. There are also links for "My Bugs", "ODR Bugs", and "Absence". The main content area displays a "Table of Contents" with the following items:

1. Listing For: Print
• [Data Set SASHELP.CLASS](#)
2. SQL is special
• [List of SASHELP.CLASS](#)
3. Listing For: Print
• [CLASS Data Set](#)
4. Listing For: Gchart
• [Student Height](#)
5. Listing For: SQL
• [List of SASHELP.CLASS](#)

The browser's status bar at the bottom indicates "Local intranet".

What are those funky characters in my HTML files?

If you ran your HTML SAS job on the mainframe, you were probably clever and created ASCII HTML via the TRANTAB option. If you did this, you can transfer the HTML files from the mainframe to some other box via FTP in binary mode. However, you probably forgot about the RECSEP option. By default, the carriage return/line feed character on the mainframe is some crazy character that only EBCDIC⁴ fans understand. RECSEP allows you to set the record separator to something more appealing to the PC and Unix boxes. See the documentation on RECSEP in *The Complete Guide to the SAS Output Delivery System* for more details. Here's an example:

```
ods html file="body.htm"
  trantab=ascii
  recsep='0D0A'x;

... sas code ...

ods html close;
```

How do I hook up ODS with SAS/Intrnet?

OK, this isn't a beginner question, but I'm going to include it anyway. No, you don't have to get rid of the SAS/Intrnet macros (if you don't know what these are, then skip to the next topic). The macros work just great. However, if you want to pipe your ODS output directly to your web browser via the application dispatcher then the ODS HTML statement needs to look like this:

```
/* Ok, so we need SAS/Intrnet. */

ods html frame=_webout (dynamic)
  file = b.html
  contents = c.html
  path = &_tmpcat (url=&_replay)
  rs=none
  trantab=ascii /* For mvs folks */
  ;

... sas code ...

ods html close;
```

The notable idea here is that the body and contents are written to catalog entries for later retrieval. The frame is piped directly to _webout. The dynamic option tells the HTML destination to write the appropriate header for the HTML. The PATH option specifies the catalog to write to. The &_tmpcat macro variable is set up by the application dispatcher, and the &_replay macro variable contains the reference that allows the links in the frame to make trips back to the server to retrieve the stored contents and body files. Got it?

⁴ IBM. Gotta love 'em.

How do I create a style of my own?

Styles can range from simple to complex. The basic idea in creating a style is to parent off of the default style in the system. In practice, this has turned out to be a bit of a chore. There is a good paper on the web that describes how to create a style from scratch. Please see <http://www.sas.com/rnd/base/topics/style-template/style.html>. Also, for more information on the mechanics of a style please see Olinger (1999).

We are currently looking at ways to simplify building a style. We hope to have something in place for release 8.1 or release 8.2.

Styles are too complex. Why can't I just specify my colors directly in my procedures?

For some procedures you can. Please see <http://www.sas.com/rnd/base/topics/v6v8ods/index.html> for a paper describing what you can achieve with the TABULATE and REPORT procedures. You can also take a look at Kelley and McNeill (1999). Lastly, for 8.2 the PRINT procedure will be able to specify stylistic elements directly as well.

You can modify style information directly on a template using the TEMPLATE procedure. This topic is a little too weighty for this paper; however, if you are interested then check out the CELLSTYLE statement and the STYLE attribute on the table and column templates. You can find this information in the TEMPLATE documentation, which is part of *The Complete Guide to the SAS Output Delivery System*.

Printer:

How do I create PDF from ODS?

This is a question that we get a lot. To answer it, I need to explain a little about the PRINTER destination:

```
ods printer;
proc print data=sashelp.class; run;
ods printer close;
```

By default, the PRINTER destination creates whatever type of output the current SAS system printer needs. For instance, if your current SAS system printer is an HP Laser Jet then you'll get output sent straight to the printer, and the only thing you will see is the hard copy on the printer.

You can set the printer via the SYSPRINT SAS option, via the SAS printer setup window, or via the ODS PRINTER sub-option. For instance, to set the printer directly in your SAS code:

```
ods printer printer="My Printer";
proc print data=sashelp.class; run;
ods printer close;
```

This example will send the PRINT procedure's results directly to the printer "My Printer". If "My Printer" is a postscript printer then you will get postscript output sent directly to the printer.

You can take this a step further by forcing ODS to write the output to a file instead of sending it directly to a printer. You achieve this with the FILE option. The following example assumes that the current system printer is a postscript printer.

```
ods printer file="print.ps";
proc print data=sashelp.class; run;
ods printer close;
```

To get PDF from this example all you would need to do is run the postscript file through Adobe Acrobat Distiller. Adobe's Distiller technology takes a postscript file and turns it into PDF. In fact, on the PC you can set Adobe Distiller as the current printer. In this case, you do not need a postscript file, as Distiller will take care of it for you.

The above example assumes that the current printer is a postscript printer. However, you can force postscript even without a postscript printer. Release 8.0 includes new "Universal Print" drivers for postscript. To use these drivers just specify the PS option:

```
ods printer file="print.ps" ps;
proc print data=sashelp.class; run;
ods printer close;
```

The PS option will force postscript output using postscript drivers written by SAS. You can then take the postscript file and distill it.

Release 8.1 will have experimental "Universal Print" drivers for PDF. What this means is that you will be able to generate PDF natively on any host⁵ without having to distill it. The PDF option will work the same as the PS option:

```
ods printer file="print.pdf" pdf;
proc print data=sashelp.class; run;
ods printer close;
```

The seminal FAQ on the printer destination is located at <http://www.sas.com/rnd/base/topics/odsprinter/index.html>. I encourage you to check it out for more information.

⁵ Any host with font support.

Can I take advantage of links and other cool PDF features?

For release 8.0, no. We support only postscript output with no way to add the appropriate information needed to make this work. However, release 8.1 supports the PDFMARK option:

```
ods printer file="print.ps" pdfmark;
proc print data=sashelp.class; run;
ods printer close;
```

The PDFMARK option turns on all the cool things that you can do with PDF via the postscript driver. With PDFMARK turned on, the PRINTER destination will create a table of contents and will recognize the URL and FLYOVER style elements.

In release 8.1, if you use the PDF option to create native PDF then PDFMARK is not necessary, as PDF will automatically handle the same functions.

As an example, consider the following:

```
data temp;
  input a b c;
  datalines;
150 45 67
97 56 44
125 55 55
;

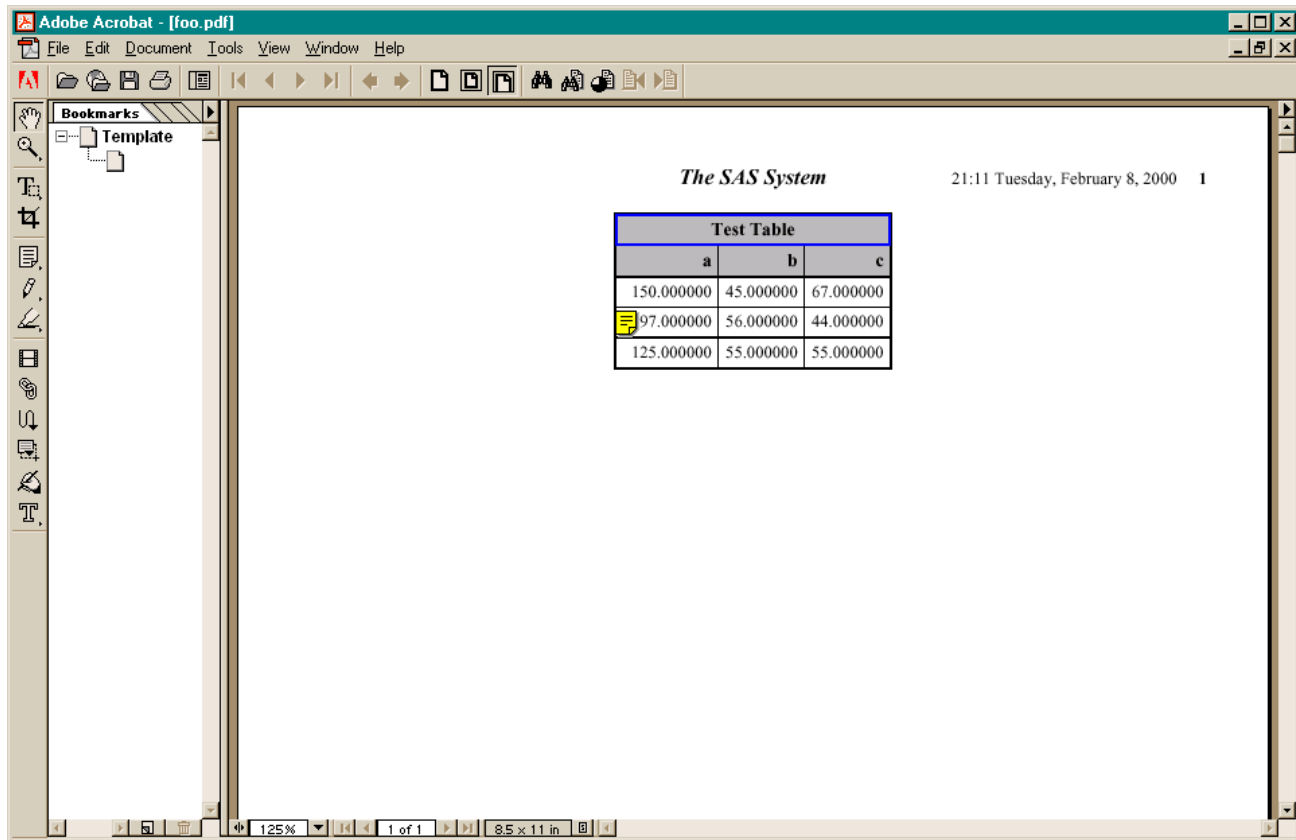
ods printer file="foo.ps" ps pdfmark;

proc template;
  define table links;
    column a b c;
    header h;
    define h;
      text "Test Table";
      style=header
        { url="http://www.sas.com" };
    end;
  define a;
    cellstyle _val_ < 100 as data
      { flyover="Too Low" };
    end;
  use_name;
end;
test data=temp;
run;

ods printer close;
```

This example gives us the output in Figure 1.

Figure 1

**RTF:****Can I create RTF output from ODS?**

This is a question that we get asked even more than “Can you produce PDF output?” The answer is “Yes, ODS can create RTF.” However, the RTF driver that is available in Release 8.0 is experimental in the broadest sense of the word. RTF is scheduled to go production in Release 8.1. This section will attempt to answer the questions that really haven’t been asked yet, but are probably going to be asked once the driver gets out there.

To create RTF output, just open the RTF destination:

```
ods rtf file="print.rtf";
proc print data=sashelp.class; run;
ods rtf close;
```

The RTF will be created and stored in print.rtf. The RTF is compatible with Word97 and above. We haven’t tried it yet with Word6, but it may work.

Caveats

To give you an idea of the difference between the Release 8.0 drivers and the Release 8.1 drivers, take a

look at the following two tables. The first table was produced with Release 8.0 and the second with Release 8.1.

Table 1

Ob s	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5

Table 2

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5

As you can see, the second table is a lot nicer. The headers are properly measured. The font is a lot nicer. The colors are a lot nicer (even if you can't see it this black and white paper). In short, it's doing things right.

So what can and what can't the RTF destination do?

Colors

The RTF destination will read an ODS style for the colors to use, but Word will only support a subset of the available colors. In fact, the colors for Word tables are restricted to a subset of sixteen fixed colors. ODS attempts to get around this by using a dithering pattern and applying two different colors to get a close approximation. The end result is that you can get a color for a cell that looks rather dimpled on the screen. However, if you print the document on a higher resolution printer, things look much better. If you stick to the basic sixteen colors no dithering is necessary.

Measuring

All tables, notes, and titles are measured horizontally. This allows us to break and wrap tables that are too wide for the page. However, we do not measure vertically because Word places items like titles and footers in unpredictable places. When a table splits across a page boundary then Word is responsible for repeating column headers. The actual measuring takes place in your SAS session and is based on a dummy printer description that has been setup to approximate what we think is a standard page description (8 1/2" by 11"). This means that you can produce RTF on any of the SAS hosts, with the same restrictions as the PRINTER destination, but if you want a strange page size then you may need to tweak the printer definition within SAS.

Horizontal-only measuring introduces some interesting artifacts. Since ODS RTF does not know where it is on the page, things like end of page totals on tables are not possible. Also, continuation notes (as in the TABULATE procedure) are not possible.

Graph

The default graph type for inclusion RTF is PNG. PNG is a non-proprietary version of GIF. You may create your graphs with whatever device you want but the final rendering will be PNG. The Graph folks are currently exploring ways to embed the ActiveX graph control within RTF. This will probably be available in Release 8.2.

URL's

RTF support URL's via the URL style element.

Bookmarks

Each piece of output gets its own bookmark. You can control the bookmark name through the BOOKMARK option.

Header Information

You can control the author, operator, and title fields in the RTF header with the corresponding RTF option.

Output Data Sets:

How do I create an output data set from ODS?

Each piece of output that is created by the procedure has attributes that you can use in your SAS code. Chief among these is the name of the output object. To discover the names of the various pieces of output you can turn on the ODS trace option:

```
ods trace output;

proc glm data=sashelp.class;
model height=weight;
quit;
```

The code above will write a trace record to the log each time a piece of output is created. A typical trace record will look like this:

```
Output Added:
-----
Name:          FitStatistics
Label:         Fit Statistics
Template:      stat.GLM.FitStatistics
Path:          GLM.ANOVA.Height.FitStatistics
-----
```

You can then use this name to turn the piece of output into an output data set using the ODS OUTPUT destination:


```
ods listing close;
ods output FitStatistics=fit;

proc glm data=sashelp.class;
  model height=weight;
quit;
```

NOTE: The data set WORK.FIT has 1 observations and 5 variables.

The output data set contains all of the raw data that made up the data component in output object. If multiple pieces of output have the same name, ODS will attempt to add all the output to the same output data set.

When I create an output data set why do I get warnings about data being dropped?

The structure of the output data set is determined by the first piece of output that matches the name on the ODS OUTPUT statement. If another piece of output with the same name is created, but with a different structure, ODS will try to put the output into the data set, but it may not be able to. If this is the case then ODS will print a warning. To solve this problem, use the MATCH_ALL option to cause each output object to go to a different data set. The MATCH_ALL option lets you specify a macro variable that will contain a list of all the data sets that were created so that you can put them back together again. This is a bit of a hack, but it does the trick. A better solution is being worked on for Release 8.2.

```
ods listing close;
ods output ModelAnova(match_all=mvar)=model;

proc glm data=sashelp.class;
  model height=weight;
quit;

data new; set &mvar; run;
```

I created a data set, but it was useless. The data wasn't in the form I needed.

Each procedure writer is responsible for ensuring that the data for a particular output object is in good shape. If this is not the case, please call tech support and report the problem. We are currently aware of a few outstanding problems that will be addressed in a forthcoming release.

Miscellaneous:

What's with PLOT and CHART? All ODS output looks like the listing!

Yes. And it will always be that way. Both PLOT and CHART and some other procedures got what is known as the "batch" treatment. Basically, we took the standard

SAS listing, captured it, and then stuffed it into whatever output destination you asked for. Batch-oriented output does not use a template and the data component of the output object is nothing more than the SAS listing stuffed into a character variable.

Can I use ODS with the Data Step?

Yes, there are 2 different ways to use ODS with the data step. You can bind a template to the data step to produce a regular table, or you can do standard data _null_ printing. If you choose the latter, be aware that all data _null_ reports will use ODS batch mode to store the tables. This is due to the fact that we really can't figure out what you are doing with your PUT statements. There really isn't a lot of semantic information available. If you use a template, however, we know that you are producing a table and we can act accordingly. For more information on how ODS interface with the data step please see Heffner (1998) and *The Complete Guide to the SAS Output Delivery System*.

We are currently working on a way for data _null_ reporting to interface with ODS on a much grander scale.

Conclusion

OK. Maybe ODS is a little ODious ☺ However, ODS is powerful system for manipulating output. With a little experience you will be able to customize output to suit your needs.

Contact Information

The SAS Institute personnel that bring you ODS are very excited about the potential of these new capabilities of the SAS System. You can send electronic mail to ods@sas.com with your comments. You can also contact the author directly at Chris.Olinger@sas.com

We will be adding more information about ODS capabilities to the Research and Development section of SAS Institute's Web Page. See <http://www.sas.com/rnd/base/>

References

Kelley, D. W. and McNeill S. L. (1999), "Getting Stylish with Version 7 Base Reporting," in the *Proceedings for the Twenty-Fourth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Heffner, W. F. (1998), "ODS: The Data Step Knows," in the *Proceedings for the Twenty-Third Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Olinger, C. R. (1999), "Twisty Little Passages, All Alike – ODS Templates Exposed" in the *Proceedings for the Twenty-Fourth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

SAS and SAS/GRAPH 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.