

## Version 9 ODS - Basics

Prepared by



### Global Business Intelligence Consulting and Training

Destiny Corporation  
100 Great Meadow Rd Suite 601  
Wethersfield, CT 06109-2379  
Phone: (860) 721-1684 1-800-7TRAINING  
Fax: (860) 721-9784  
Email: [info@destinycorp.com](mailto:info@destinycorp.com)  
Web: [www.destinycorp.com](http://www.destinycorp.com)

#### Overview of ODS

Procedural output may be routed to:

- A Data Set -- using a Data Step
- A Flat File -- using Proc printto
- A Graph -- using Proc gchart
- The Output Window -- default for Proc print

Programming capabilities include the use of Proc printto to direct Proc print output to a flat file. The Data\_Null\_Step allows the Put statement to direct output to files without creating a data set.

These options are still viable and will continue to be used.

With ODS, additional programming syntax provides variety and strength for displaying results.

#### Creating ODS Output

In SAS, procedural output may be routed to a variety of different destinations.

The current list includes the following:

- Listing Window
- HTML Documents
- Output Data Sets
- Graphic Streams
- Active X
- Java
- Rich Text Format (\*.Rtf) Files
- Printer Locations
- Postscript And PCL (For High Fidelity Printers)
- SGML And Latex (Tentative)

In addition to sending output to a variety of new destinations, the appearance of the output can be modified with different fonts, colors and formatting.

To understand how SAS achieves these results and to get acquainted with the syntax of ODS, consider the following excerpt from the SAS Institute reference material:

"All SAS procedures produce *output objects* that the Output Delivery System delivers to various ODS

*destinations*, according to the default specifications for the procedure or to your own specification".

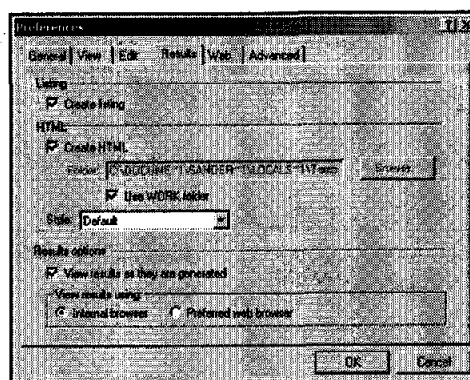
All output objects (for example, a table of parameter estimates) consist of two component parts:

- The data component, which consists of the results computed by a SAS procedure.
- The template, which contains rules for formatting and displaying the results.

The familiar part of the ODS is that output is channeled to the Listing window (the Output window).

To assure that output is written to the Listing window, select Tools → Options → Preferences from the pull-down menu in the Program Editor.

A Preferences window appears with a Results tab.



The Results tab permits SAS output to be sent to the Listing window, to the Results Viewer (with HTML content), to both the Listing and Results Viewer window, or to none of the above.

The Listing window is open by default.

SAS Institute has also created a series of templates with distinct colors, fonts, and HTML options enhancing the appearance of the file in the browser.

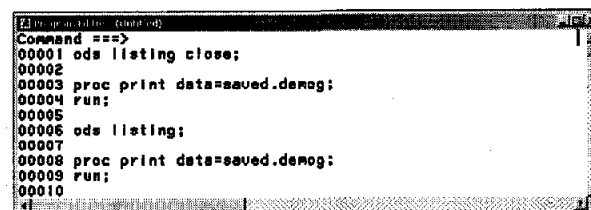
The styles include Beige, Brick, Brown, D3D, Default, Minimal, Printer, and Statdoc in Version 8.

The additional styles of BarrettsBlue, FancyPrinter, NoFontPrinter, Rtf, SansPrinter, and SerifPrinter were added in Version 8e.

#### ODS Listing

Assume that the Results tab in the Preferences window has both the Listing window and HTML output open.

For this exercise, we wish to publish only HTML output. Closing the Listing window will increase efficiency.



Be careful to have at least the Listing window or the Results window open.

If both windows are deactivated, the Log will display a Warning message that no output destinations are active.

```

Command ==>
33 ods listing close;
34
35 proc print data=saved.demog;
36 run;

WARNING: No output destinations active.
NOTE: PROCEDURE PRINT used:
      real time    0.01 seconds
      cpu time     0.01 seconds

```

## Introduction to Output in ODS Terms

Consider the HTML output for Proc Contents.

```

Command ==>
00001 proc sort data=saved.demog out=work.demog;
00002   by gender status;
00003 run;
00004
00005 proc sql;
00006   create index money on work.demog (salary, children);
00007 quit;
00008
00009 ods listing close;
00010 proc contents data=work.demog;
00011 run;
00012 ods listing;
00013

```

The Results window shows that Proc Contents has multiple separate parts.

Each part of the output is a separate HTML table.

Other Proc output is actually composed of multiple tables.

| HEIGHT | Num  | 9  | 8  |
|--------|------|----|----|
| NAME   | Char | 25 | 48 |
| SALARY | Num  | 9  | 40 |
| STATUS | Char | 9  | 81 |
| WEIGHT | Num  | 8  | 16 |

| SALARY CHILDREN |    |
|-----------------|----|
| MONEY           | 47 |

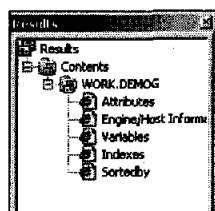
  

| GENDER STATUS |      |
|---------------|------|
| VARIABLE      | YES  |
| Character Set | ANSI |

## Components of ODS Output

In many cases, separate HTML tables combine to create one Proc output. Why is this important?

First, consider the Results window, which allows the user to navigate directly to a specific HTML table by clicking on its name.



Although each of the tables is contained in a single HTML file, navigation is made possible to any table through hyperlinks in the HTML file.

The HTML hyperlinks providing this navigation option are shown below.

By asking to inspect the source code, the hyperlinks are made visible. Hyperlinks can be identified by the syntax <A NAME="IDX3">, or where the number is any number. In the example below, IDX3 and IDX4, are both shown.

```

<TABLE>
</TABLE>
<A NAME="IDX3">&nbsp;&nbsp;&nbsp;</A>
<CENTER>
<font face="Arial, Helvetica, Helv" size="3" color="#002288">
<TABLE cellpadding=7 rules=groups frame=box border=1 bgcolor="#f0f0f0" border
<thead>
<tr>
<td colspan=4 align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, H
</tr>
<tr>
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size=
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size=4
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size=
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size=4
</tr>
</thead>
<tbody>
<tr>
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size="4" color="#002
<td align=left align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size="3" color="#0000
<td align=right align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size="3" color="#0000
<td align=left align=bottom bgcolor="#f0f0f0"><font face="Arial, Helvetica, Helv" size="3" color="#0000
</tr>
</tbody>
</table>
</font></center>
<A NAME="IDX4">&nbsp;&nbsp;&nbsp;</A>

```

These hyperlinks are part of the document. They are created automatically when the HTML destination is open.

We can customize their syntax to take advantage of them for navigation purposes.

## ODS Trace

There is a second reason for examining HTML output as separate tables: the programmer may wish to show only selected portions of output without showing other parts.

ODS programming allows the specification of which tables to suppress or present.

Suppressing unwanted parts of the output increases efficiency.

To find out more about the components of the Proc output, use the ODS trace statement.

```

Command ==>
00001 ods trace on;
00002   proc contents data=work.demog;
00003   run;
00004 ods trace off;
00005

```

This syntax requests information into the Log, which is essential for using ODS options.

The Log will now show information about each table.

```

1815 ods trace on;
1816 proc contents data=work.demog;
1817 run;

NOTE: Writing HTML Body File: asent8.htm

Output Added:
Name: Attributes
Label: Attributes
Template: Base.Contents.Attributes
Path: Contents.Dataset.Attributes

Output Added:
Name: EngineHost
Label: Engine/Host Information
Template: Base.Contents.EngineHost
Path: Contents.Dataset.EngineHost

Output Added:
Name: VariableAlpha
Label: Variable
Template: Base.Contents.Variable
Path: Contents.Dataset.VariableAlpha

Output Added:
Name: Indexes
Label: Indexes
Template: Base.Contents.Indexes
Path: Contents.Dataset.Indexes

Output Added:
Name: Sortedby
Label: Sortedby
Template: Base.Contents.Sortedby
Path: Contents.Dataset.Sortedby

```

By default, the trace record contains the following information:

| Item      | Description                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------|
| Name      | Name of the output object                                                                                          |
| Label     | Brief description of the output object                                                                             |
| Data Name | Name of data component used to create the output; provided only when it differs from the name of the output object |
| Template  | Template name used to create output                                                                                |
| Path      | Path of the output object                                                                                          |

### ODS Select and ODS Exclude

Suppose we were interested solely in the information about the variables and the indexes.

The syntax should request information be displayed on the selected tables only (or the syntax should specifically exclude the other tables).

```

Command ==>
00001 title "Full Output of Proc Contents";
00002 proc contents data=work.demog;
00003 run;
00004
00005 title "Partial Output of Proc Contents";
00006 ods select contents.dataset.variablesalpha
00007         contents.dataset.indexes;
00008 proc contents data=work.demog;
00009 run;
00010
00011 title;
00012

```

The output of the second Proc Contents is shown below. The output shows only the variables and the indexes.

```

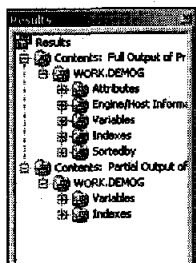
1341 title "Full Output of Proc Contents";
1342 proc contents data=work.demog;
1343 run;
NOTE: Writing HTML Body file: eashnm13.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.22 seconds
      cpu time     0.06 seconds

1344 title "Partial Output of Proc Contents";
1345 ods select contents.dataset.variablesalpha
1346         contents.dataset.indexes;
1347 proc contents data=work.demog;
1348 run;
NOTE: Writing HTML Body file: eashnm14.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.17 seconds
      cpu time     0.04 seconds

1350
1351 title;

```

The Results window shows the full output and the partial output.



The same output is created using the Exclude action.

```

Command ==>
00001 ods exclude contents.dataset.attributes
00002         contents.dataset.enginehost
00003         contents.dataset.sortedby;
00004 proc contents data=work.demog;
00005 run;
00006

```

Both the Select action and the Exclude action can be used to subset the output.

Since output often consists of multiple tables, the programmer can specify which portions of a Proc will be displayed.

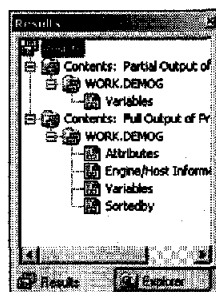
The Select action and Exclude action are in effect for the Proc, which immediately follows them.

```

Command ==>
00001 title "Partial Output of Proc Contents";
00002 ods exclude contents.dataset.attributes
00003         contents.dataset.enginehost
00004         contents.dataset.sortedby;
00005 proc contents data=work.demog;
00006 run;
00007
00008 title "Full Output of Proc Contents";
00009 proc contents data=work.demog;
00010 run;
00011 title;

```

The Results window shows the differences in output.



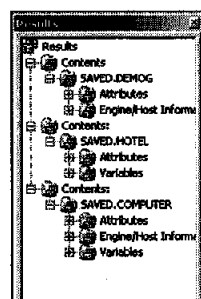
In order to extend the functionality of the Select action or the Exclude action to the next procedural output, use the Persist syntax.

```

Command ==>
00001 /* predict the output for each Proc Contents */
00002
00003 ods listing;
00004 ods listing show;
00005 ods exclude variablesalpha;
00006 proc contents data=saved.demog;
00007 run;
00008 ods listing show;
00009 ods exclude enginehost;
00010 proc contents data=saved.hotel;
00011 run;
00012 ods listing show;
00013 ods exclude none;
00014 proc contents data=saved.computer;
00015 run;
00016 ods listing show;
00017 ods listing close;
00018

```

The Results window will display the following.



```

1004 ods listing;
1005 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is:
1: variableAlpha(PERSIST)
1006 ods exclude variableAlpha;
1007 proc contents data=saved.demog;
1008 run;

NOTE: Writing HTML Body File: aschta14.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.22 seconds
      cpu time     0.03 seconds

1009 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is: ALL
1010 ods exclude enginehost;
1011 proc contents data=saved.hotel;
1012 run;

NOTE: Writing HTML Body File: aschta15.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.19 seconds
      cpu time     0.03 seconds

1013 ods listing show;
Current LISTING select list is set to default value (ALL).
1014 ods exclude none;
1015 proc contents data=saved.computer;
1016 run;

NOTE: Writing HTML Body File: aschta16.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.21 seconds
      cpu time     0.03 seconds

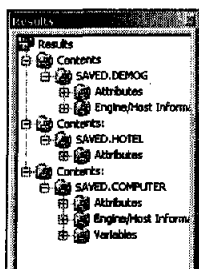
1017 ods listing show;
Current LISTING select list is set to default value (ALL).
1018 ods listing close;
1019

```

```

Command ===>
00019 /* explain the output for each Proc Contents*/
00020 ods listing;
00021 ods listing show;
00022 ods exclude variableAlpha (persist);
00023 proc contents data=saved.demog;
00024 run;
00025 ods listing show;
00026 ods exclude enginehost (persist);
00027 proc contents data=saved.hotel;
00028 run;
00029 ods listing show;
00030 ods exclude none;
00031 proc contents data=saved.computer;
00032 run;
00033 ods listing show;
00034 ods listing close;
00035

```



```

1020 ods listing;
1021 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is:
1: variableAlpha(PERSIST)
1022 ods exclude enginehost (persist);
1023 proc contents data=saved.hotel;
1024 run;

NOTE: Writing HTML Body File: aschta17.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.22 seconds
      cpu time     0.03 seconds

1025 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is:
1: variableAlpha(PERSIST)
1026 ods exclude none;
1027 proc contents data=saved.computer;
1028 run;

NOTE: Writing HTML Body File: aschta18.htm
NOTE: PROCEDURE CONTENTS used:
      real time    0.22 seconds
      cpu time     0.03 seconds

1029 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is: NONE
1030 ods listing close;
1031

```

To review, examine the following variations on ODS Select and ODS Exclude syntax.

```

ods select <table_template_name>;
ods select all;
ods select;
ods select <table_template_name>
(persist);

ods exclude <table_template_name>;
ods exclude none;
ods exclude;
ods exclude <table_template_name>
(persist);

```

The default values depend on the ODS destination:

- ODS LISTING: select all
- ODS HTML: select all
- ODS OUTPUT: exclude all
- Overall: select all

### Using Labels and Listing Options on ODS Trace Statements

The ODS trace statement uses the Label and Listing options, as displayed.

```

1424 ods trace on / label;
1425 proc contents data=work.demog;
1426 run;

Output Added:
Name: Attributes
Label: Attributes
Template: Base.Contents.Attributes
Path: Contents.DataSet.Attributes
Label Path: "The Contents Procedure"."WORK.DEMOG"."Attributes"

```

The Label option displays additional information in the Log window.

The Listing option includes the trace details in the Listing window along with the Proc contents output.

```

Data Set Page Size: 12288
Number of Data Set Pages: 1
First Data Page: 1
Max Obs per Page: 117
Obs in First Data Page: 48
Number of Data Set Repetitions: 1
File Name: C:\DOCUMENT\1\SASWORK\1\LOCALS\1\Temp\SSR
Temporary Files\_T0205\demog.aschta18
Release Created: 8.0.1010
Host Created:

Output Added:
Name: VariableAlpha
Label: Variable
Template: Base.Contents.VariableAlpha
Path: Contents.DataSet.VariableAlpha

```

The Listing option and the Label option may be used together. When Listing is used with or without Label, the Log window does not contain the trace details.

### Alternative Table References

The ODS trace statement showed the following features of each output table:

- Name
- Label
- Template
- Path
- Label Path

The syntax shown earlier, for the Select action and the Exclude action, referenced the full path.

We can use syntax for the Select action and Exclude action to reference any of the following parts of a table's features:

- Full path
- Partially qualified path
- Label (surrounded by quotation marks)
- Label path
- Partial label path
- Mixture of labels and paths
- Any above specification with # plus number

A partially qualified path is truncated by removing sections from the left side up to each dot.

For example, here is a full path:

contents.datasets.indexes

The above path may be referenced by the partial path designations displayed below:

datasets.indexes  
indexes

Partially qualified label paths observe the same truncation rule of removing sections from the left side up to each dot.

The following syntax displays different ways of referencing output. The same output table has been referenced in different ways.

```
Command ==>
00001 ods trace on / label;
00002 proc contents data=saved.demog;
00003 run;
00004 ods trace off;
00005
```

```
NOTE: Writing HTML Body file: sashtm2.htm
Output Added:
Name: All Variables
Label: All Variables
Path: Contents.DataSet.EngineHost
Label Path: "The Contents Procedure". "Saved.Demog". "Engine/Host Information"
Output Added:
Name: Engine/Host Information
Label: Engine/Host Information
Path: Contents.DataSet.EngineHost
Label Path: "The Contents Procedure". "Saved.Demog". "Engine/Host Information"
Output Added:
Name: Variables
Label: Variables
Path: Contents.DataSet.EngineHost
Label Path: "The Contents Procedure". "Saved.Demog". "Variables"
NOTE: PROCEDURE CONTENTS used:
real time 0.18 seconds
cpu time 0.18 seconds
1173 ods trace off;
```

| EngineHost                                                               | Name               |
|--------------------------------------------------------------------------|--------------------|
| "Engine/Host Information"                                                | Label              |
| Contents.DataSet.EngineHost                                              | Path               |
| DataSet.EngineHost                                                       | Partial Path       |
| EngineHost                                                               | Partial Path       |
| "The Contents Procedure".<br>"Saved.Demog". "Engine/Host<br>Information" | Label Path         |
| "Saved.Demog". "Engine/Host<br>Information"                              | Partial Label Path |
| "Engine/Host Information"                                                | Partial Label Path |

A combination of labels and paths can be used.

Any path specification followed by a # and a number will reference that name and the numbered output.

The references above are not case sensitive.

Of the information in the Trace output, only the Template information is not used in the Select and Exclude syntax.

## ODS Listing Syntax

Although the Listing window is open by default, it can be opened, managed, and closed as needed.

- ODS LISTING
- ODS LISTING EXCLUDE
- ODS LISTING SELECT
- ODS LISTING SHOW
- ODS LISTING CLOSE

```
Command ==>
00001 ods listing exclude variables=alpha;
00002 proc contents data=saved.demog;
00003 run;
00004
00005 ods listing select attributes;
00006 proc contents data=saved.computer;
00007 run;
00008
```

The ODS Listing statement with 'show' can be helpful in determining subset information.

```
Command ==>
00001 ods listing show;
00002 ods listing exclude variables=alpha;
00003 ods listing show;
00004 proc contents data=saved.demog;
00005 run;
00006
00007 ods listing show;
00008 proc contents data=saved.computer;
00009 run;
00010 ods listing close;
00011
```

```
1470 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is: ALL
1471 ods listing exclude variables=alpha;
Current LISTING exclude list is:
1. variables=alpha
1472 proc contents data=saved.demog;
1473 run;
NOTE: Writing HTML Body file: sashtm4.htm
NOTE: PROCEDURE CONTENTS used:
real time 0.56 seconds
cpu time 0.04 seconds
1474
1475 ods listing show;
Current LISTING select list is set to default value (ALL).
Current OVERALL select list is: ALL
1476 proc contents data=saved.computer;
1477 run;
NOTE: Writing HTML Body file: sashtm5.htm
NOTE: PROCEDURE CONTENTS used:
real time 0.20 seconds
cpu time 0.03 seconds
1478 ods listing close;
```

Note: The reference to variables=alpha in Proc contents is correct for version 8. Beginning with version 9, the preferred term is variables.

## Enhancing ODS HTML Output

We can enhance ODS HTML output using the ODS HTML File =option.

```
Command ==>
00001 ods html file="a:\myoutput.html";
00002 proc print data=saved.demog (obs=3);
00003 var name age salary children;
00004 run;
00005 ods html close;
00006
```

Aspects of this syntax include the following:

- The file extension must be either 'html' or 'htm'
- The file= syntax can also use body=
- Multiple procedures can be included

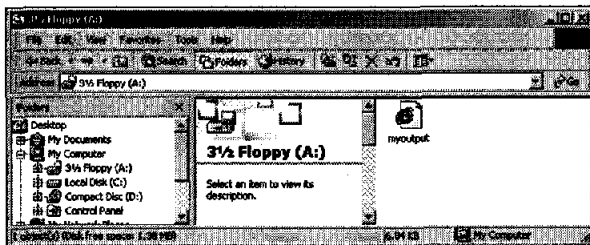
```

718 ods html file="a:\myoutput.html";
NOTE: Writing HTML Body file: a:\myoutput.html
719 proc print data=saved.demog (obs=3);
720 var name age salary children;
721 run;

NOTE: There were 3 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE PRINT used:
      real time      1.08 seconds
      cpu time        0.01 seconds

722 ods html close;

```



| NAME            | AGE | SALARY | CHILDREN |
|-----------------|-----|--------|----------|
| Carol Dwyer     | 22  | 15582  | 0        |
| Julia Pendebury | 26  | 8870   | 1        |
| Norman Harvey   | 29  | 12673  | 2        |

Enhancing ODS HTML Output / File=, Contents=

```

Command ==>

00001 ods listing close;
00002 ods html file="a:\myoutput.html"
00003 contents="a:\mycontents.html";
00004 proc print data=saved.demog;
00005 run;
00006 proc contents data=saved.demog;
00007 run;
00008 proc univariate data=saved.demog;
00009 run;
00010 ods html close;
00011 ods listing;
00012

```

```

751 ods listing close;
752 ods html file="a:\myoutput.html";
753 contents="a:\mycontents.html";
NOTE: Writing HTML Body file: a:\myoutput.html
NOTE: Writing HTML Contents file: a:\mycontents.html
754 proc print data=saved.demog;
755 run;

NOTE: There were 50 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE PRINT used:
      real time      4.86 seconds
      cpu time        0.82 seconds

756 proc contents data=saved.demog;
757 run;

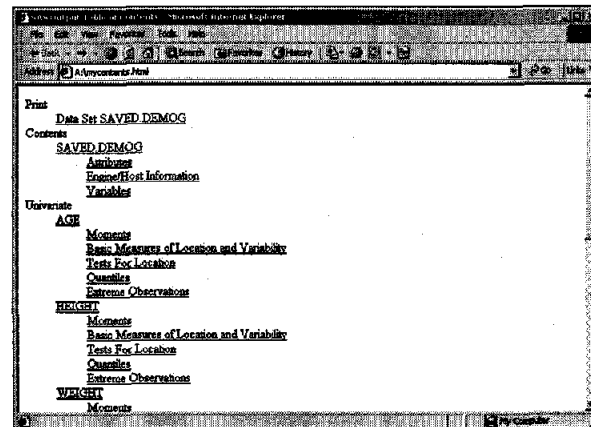
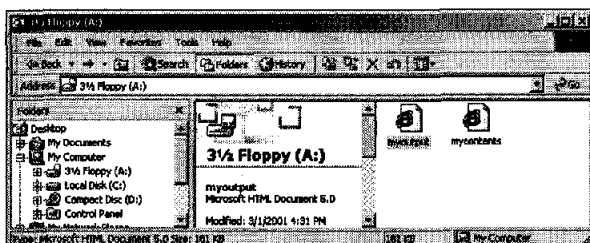
NOTE: PROCEDURE CONTENTS used:
      real time      2.38 seconds
      cpu time        0.59 seconds

758 proc univariate data=saved.demog;
759 run;

NOTE: There were 50 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE UNIVARIATE used:
      real time      6.77 seconds
      cpu time       1.15 seconds

760 ods html close;
761 ods listing;

```



| Extreme Observations |        |     |         |     |
|----------------------|--------|-----|---------|-----|
|                      | Lowest |     | Highest |     |
|                      | Value  | Obs | Value   | Obs |
|                      | 52     | 37  | 87.0    | 2   |
|                      | 52     | 32  | 87.0    | 6   |
|                      | 52     | 30  | 87.0    | 14  |
|                      | 52     | 9   | 87.0    | 39  |
|                      | 53     | 34  | 88.5    | 11  |

Enhancing ODS HTML Output / File=, Page=

```

Command ==>

00001 ods listing close;
00002 ods html file="a:\myoutput.html"
00003 page="a:\mypages.html";
00004 proc print data=saved.demog;
00005 run;
00006 proc contents data=saved.demog;
00007 run;
00008 proc univariate data=saved.demog;
00009 run;
00010 ods html close;
00011 ods listing;
00012

```

```

773 ods listing close;
774 ods html file="a:\myoutput.html"
775 page="a:\mypages.html";
NOTE: Writing HTML Body file: a:\myoutput.html
NOTE: Writing HTML Page file: a:\mypages.html
776 proc print data=saved.demog;
777 run;

NOTE: There were 50 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE PRINT used:
      real time      2.58 seconds
      cpu time        0.63 seconds

778 proc contents data=saved.demog;
779 run;

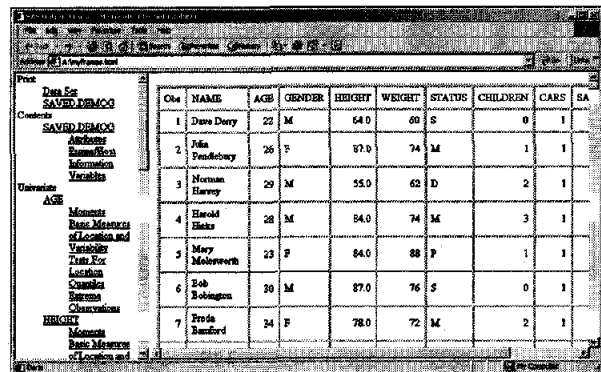
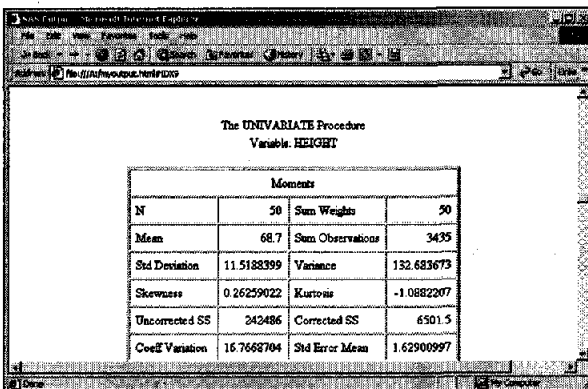
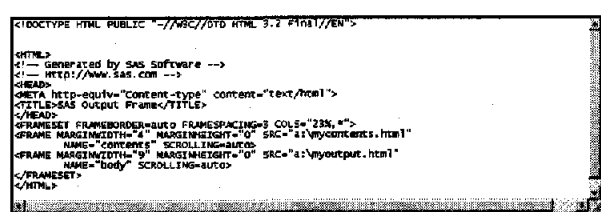
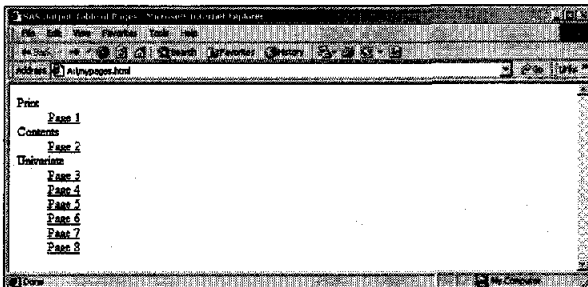
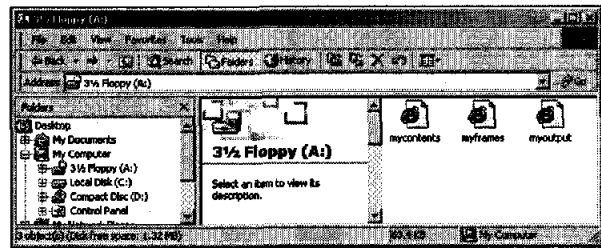
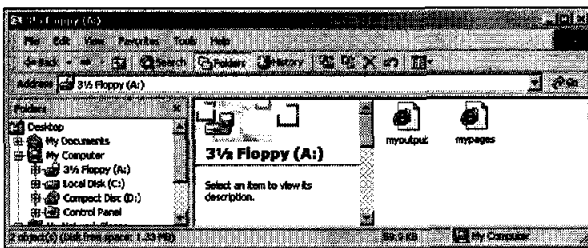
NOTE: PROCEDURE CONTENTS used:
      real time      1.34 seconds
      cpu time        0.02 seconds

780 proc univariate data=saved.demog;
781 run;

NOTE: Unable to find the "Note" style element. Default style attributes will be used.
NOTE: There were 50 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE UNIVARIATE used:
      real time      3.56 seconds
      cpu time       0.15 seconds

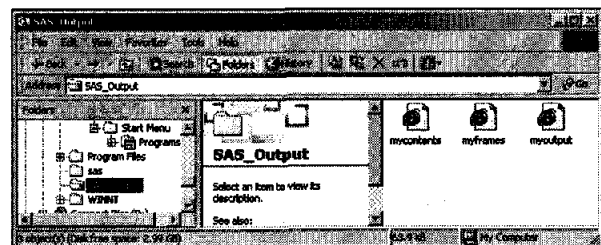
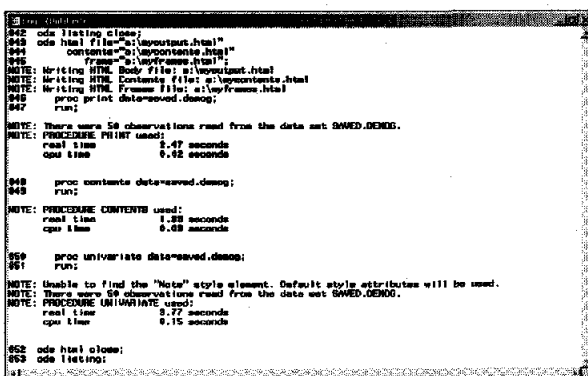
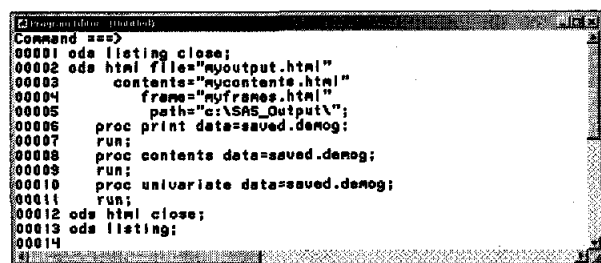
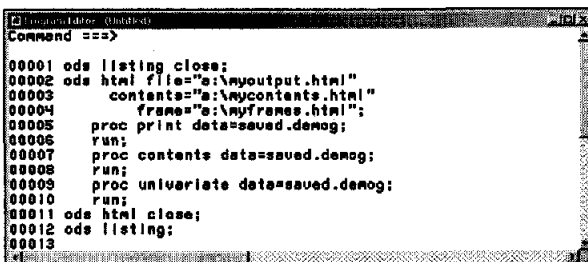
782 ods html close;
783 ods listing;

```



## Enhancing ODS HTML Output / File=, Contents=, Frame=, Path=

### Enhancing ODS HTML Output / File=, Contents=, Frame=



## Creating Hyperlinks – Path, Base and URL Syntax

The hyperlink paths created in the Frames file are important.

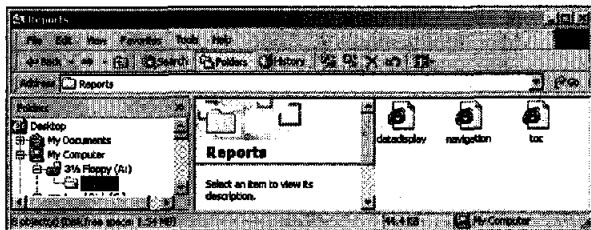
Correct use of the URL=, Path= and Base= options assure desired results.

The first syntax to consider is omitting the URL reference entirely.

```

Command ==>
00001 ods html body="a:\reports\datadisplay.html"
00002 contents="a:\reports\toc.html"
00003 frame="a:\reports\navigation.html";
00004 proc univariate data=saved.demog;
00005 run;
00006 ods html close;
00007
nnnnn

```



Notice the SRC= syntax which provides path information referencing the drive and subfolders containing the file.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>
<!-- Generated by SAS Software -->
<!-- http://www.sas.com -->
<HEAD>
<META http-equiv="Content-type" content="text/html">
<TITLE>SAS Output Frame</TITLE>
</HEAD>
<FRAMESET FRAMEBORDER=auto FRAMESPACING=3 COLS="23%,*">
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0"
SRC="a:\reports\toc.html" NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="0" MARGINHEIGHT="0"
SRC="a:\reports\datadisplay.html" NAME="body" SCROLLING=auto>
</FRAMESET>
</HTML>

```

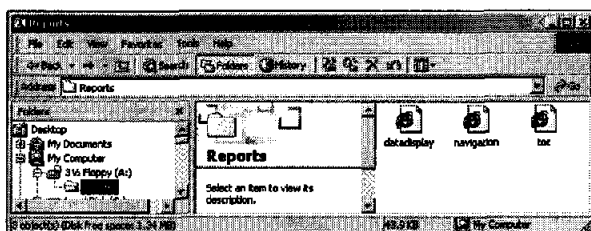
Consider using the PATH= and URL=NONE explicitly stated in the syntax.

In this syntax, the hyperlinks lack information on the drive and folder. Only the file is referenced. It implies that the file must be in the same disk location as the frames file.

```

Command ==>
00001 ods html path = "a:\reports" (url=none)
00002 body = "datadisplay.html"
00003 contents = "toc.html"
00004 frame = "navigation.html";
00005 proc univariate data=saved.demog;
00006 run;
00007 ods html close;
00008
nnnnn

```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>
<!-- Generated by SAS Software -->
<!-- http://www.sas.com -->
<HEAD>
<META http-equiv="Content-type" content="text/html">
<TITLE>SAS Output Frame</TITLE>
</HEAD>
<FRAMESET FRAMEBORDER=auto FRAMESPACING=3 COLS="23%,*">
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0"
SRC="toc.html" NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="0" MARGINHEIGHT="0"
SRC="datadisplay.html" NAME="body" SCROLLING=auto>
</FRAMESET>
</HTML>

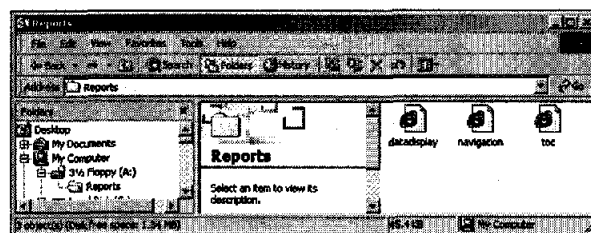
```

File location can be seen when supplying the Path=, URL=NONE, and Base= information.

```

Command ==>
00001 ods html path="a:\reports" (url=none)
00002 base="http://www.destinycorp.com/data/"
00003 body="datadisplay.html"
00004 contents="toc.html"
00005 frame="navigation.html";
00006 proc univariate data=saved.demog;
00007 run;
00008 ods html close;
00009
nnnnn

```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

<HTML>
<!-- Generated by SAS Software -->
<!-- http://www.sas.com -->
<HEAD>
<META http-equiv="Content-type" content="text/html">
<TITLE>SAS Output Frame</TITLE>
</HEAD>
<FRAMESET FRAMEBORDER=auto FRAMESPACING=3 COLS="23%,*">
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0"
SRC="http://www.destinycorp.com/data/toc.html" NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="0" MARGINHEIGHT="0"
SRC="http://www.destinycorp.com/data/datadisplay.html" NAME="body" SCROLLING=auto>
</FRAMESET>
</HTML>

```

### Titles and Footnotes in HTML Output/Report Styles

Titles and footnotes have always been available for procedural output. With ODS HTML output, titles and footnotes can take on the new role of hyperlinks.

Examine the following output. It might be helpful to have navigation at the top and/or bottom of the browser page.

```

Command ==>
00001 ods html title = 'a:\freq.html' style=brick;
00002 title 'a href = "a:\target.html"
00003 Return to target </a>';
00004 proc freq data = saved.demog;
00005 table gender;
00006 run;
00007 ods html close;
00008
nnnnn
00010
00011 ods html file = 'a:\target.html' style=brick;
00012 footnote 'a href = "a:\freq.html"
00013 Return to homepage </a>';
00014 proc print data = saved.demog (where = (gender = 'F'));
00015 run;
00016 proc print data = saved.demog (where = (gender = 'M'));
00017 run;
00018 ods html close;
00019
nnnnn

```

Note the title statement in the first ods section includes an href to the second ods section. Likewise the second ods has an href in the footnote statement to the first ods. By clicking on either underlined reference, the viewer presents the other html page.

**Return to target**

**The FREQ Procedure**

|   | Frequency | Percent | Cumulative Frequency | Cumulative Percent |
|---|-----------|---------|----------------------|--------------------|
| F | 22        | 22.00   | 22                   | 22.00              |
| M | 28        | 28.00   | 50                   | 50.00              |

| Obs | NAME | AGE | GENDER | HEIGHT | WEIGHT | STATUS | CHILDREN | CAR |
|-----|------|-----|--------|--------|--------|--------|----------|-----|
| 2   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 3   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 5   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 7   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 8   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 9   | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 10  | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |

By scrolling to the bottom of the female (or male) gender observations, an href allows a return to the other page with a click.

| Obs                                | NAME | AGE | GENDER | HEIGHT | WEIGHT | STATUS | CHILDREN | CAR |
|------------------------------------|------|-----|--------|--------|--------|--------|----------|-----|
| 39                                 | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| 40                                 | ...  | ... | ...    | ...    | ...    | ...    | ...      | ... |
| <a href="#">Return to homepage</a> |      |     |        |        |        |        |          |     |

### ODS HTML <...> NEWFILE =

By default, all output is put into a single Body file.

This can be expressed in syntax with the NEWFILE=NONE option.

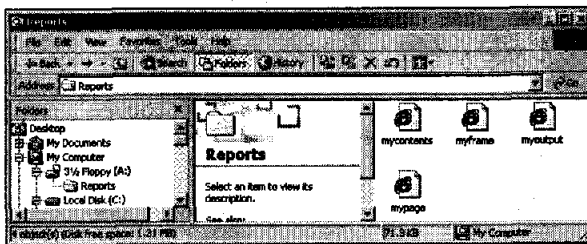
In contrast, multiple documents can be created, each holding separate procedural output. If the Body file ends in a number, procedural output is put into separate files, each numbered from the original starting point.

A new file of output can also be created with each page of output (NEWFILE=PAGE) or each Proc output (NEWFILE=PROC). A separate file can also be created for each piece of output (NEWFILE=OUTPUT).

```

Command ==>
00001 ods html body="myoutput.html"
00002 contents="mycontents.html"
00003 page="mypage.html"
00004 frame="myframe.html"
00005 path="a:\Reports"
00006 newfile=proc;
00007 proc print data=saved.demog;
00008 run;
00009 proc contents data=saved.demog;
00010 run;
00011 proc univariate data=saved.demog;
00012 run;
00013 ods html close;
00014

```

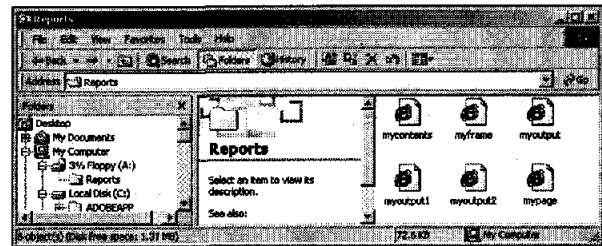


Create separate files for each procedural output.

```

Command ==>
00001 ods html body="myoutput.html"
00002 contents="mycontents.html"
00003 page="mypage.html"
00004 frame="myframe.html"
00005 path="a:\Reports"
00006 newfile=proc;
00007 proc print data=saved.demog;
00008 run;
00009 proc contents data=saved.demog;
00010 run;
00011 proc univariate data=saved.demog;
00012 run;
00013 ods html close;
00014

```



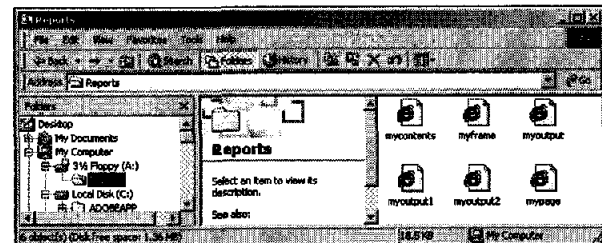
| The CONTENTS Procedure |                              |                       |     |
|------------------------|------------------------------|-----------------------|-----|
| Data Set Name:         | SAVED.DEMOG                  | Observations:         | 50  |
| Member Type:           | DATA                         | Variables:            | 10  |
| Engine:                | V8                           | Indexes:              | 0   |
| Created:               | 8:38 Monday, October 4, 1999 | Observation Length:   | 104 |
| Last Modified:         | 8:38 Monday, October 4, 1999 | Deleted Observations: | 0   |
| Protection:            |                              | Compressed:           | NO  |
| Data Set Type:         |                              | Sorted:               | NO  |
| Label:                 |                              |                       |     |

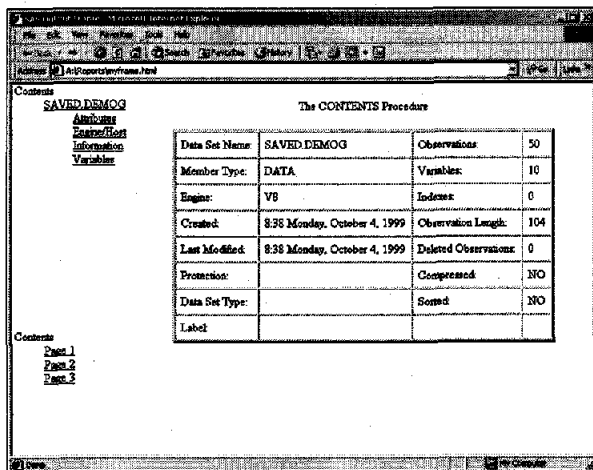
Create separate files for each table template of the document.

```

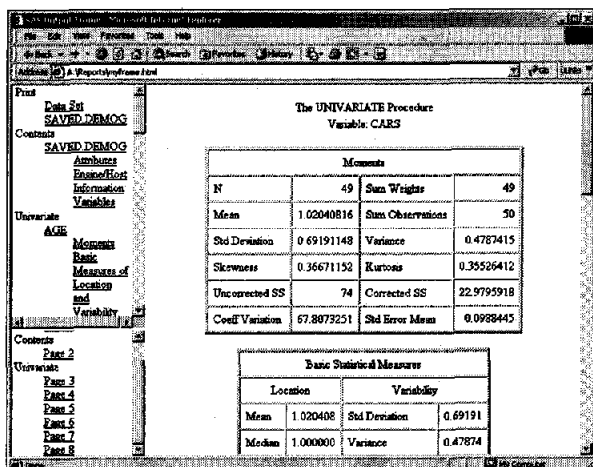
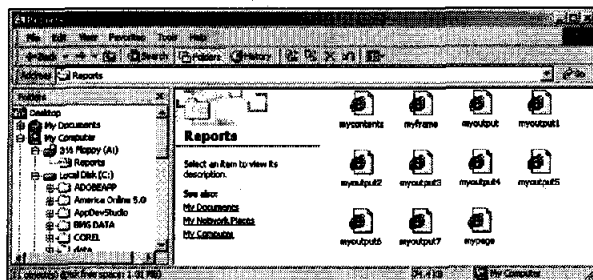
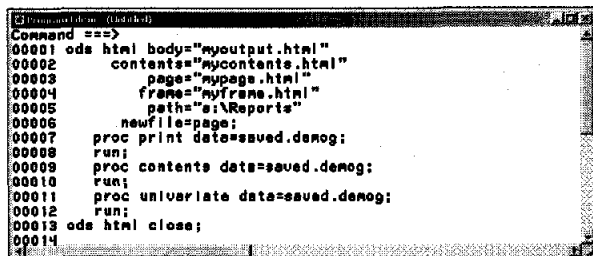
Command ==>
00001 ods html body="myoutput.html"
00002 contents="mycontents.html"
00003 page="mypage.html"
00004 frame="myframe.html"
00005 path="a:\Reports"
00006 newfile=output;
00007 proc contents data=saved.demog;
00008 run;
00009 ods html close;
00010

```



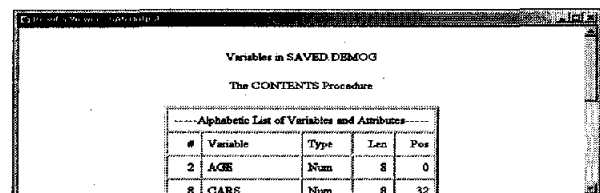
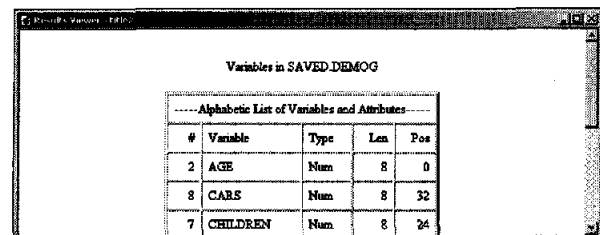
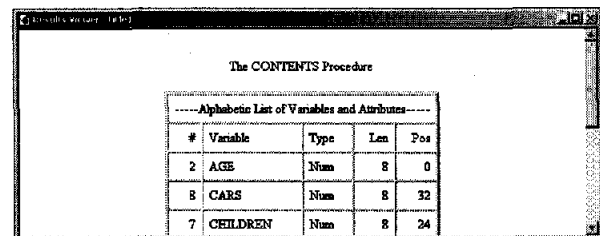
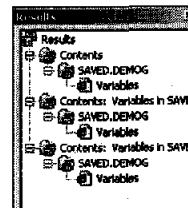
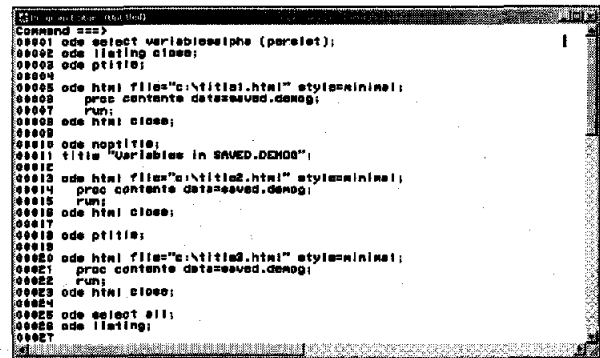


Create a new file for each page output of the procedure.



## ODS NOPTITLE

To remove the title of the Procedure, use ODS Noptitle.



## ODS HTML with Graphs – Introduction

SAS can create graphic output (with the SAS/GRAPH license) directly to the Web.

This process uses Web drivers:

- GIF
- ACTIVE X
- JAVA
- WEBFRAME
- GIFANIM
- JAVAMET (v. 8.1)

The SAS/GRAPH procedures that may be used with Gif, Active X and Java include:

- GANNO (Gif)
- GCHART (Gif, Active X, Java)
- GCONTOUR (Gif, Active X, Java)
- GFONT (Gif)
- GIMPORT (Gif)
- GMAP (Gif, Active X, Java)
- GPLOT (Gif, Active X, Java)
- GPRINT (Gif)
- GREPLAY (Gif)
- GTESTIT (Gif)
- GSLIDE (Gif)
- G3D (Gif, Active X, Java)
- G3GRID (Gif)

Not all graph options can be applied to each output destination. Check your documentation for specifics and version updates.

#### Active X

Version 8 onwards, SAS/GRAPH can create Active X graphs for viewing in Web browser mode.

Once the graph is created, it can be customized without having to re-run it.

For example, a vertical bar chart can be changed into a horizontal bar chart or a pie chart. Features such as type, color and axes can be modified.

Features of SAS/GRAPH procedures that are not supported include:

- User-defined formats
- Fonts (other than certain PC TrueType fonts)

Active X does not support the following procedures:

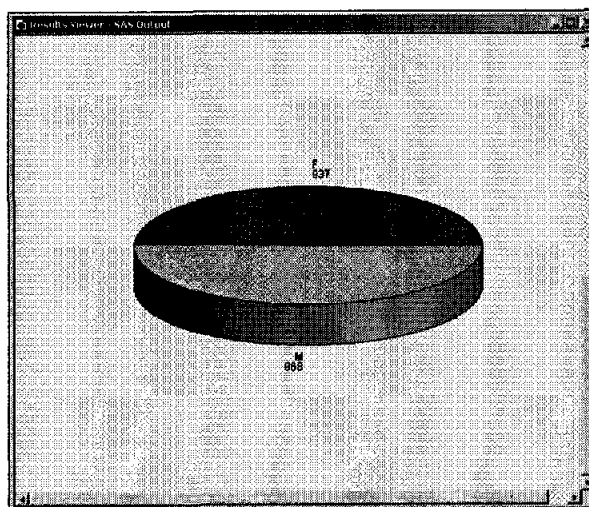
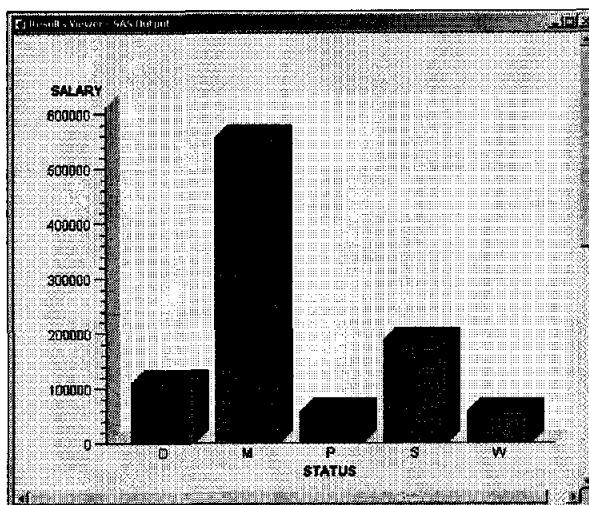
- GTESTIT
- GSLIDE
- GREPLAY
- GANNO
- GPRINT

It will produce Active X output.

```

Command ==>
00001 goptions reset=global
00002          reset=all
00003          device = activex transparency;
00004
00005 ods html file="c:\myglf.html";
00006 proc gchart data=sasuser.demog;
00007   ubar3d status / sumvar = salary;
00008 run;
00009 pie3d gender / sumvar = age;
00010 run;
00011 quit;
00012 ods html close;
00013

```



#### Java

SAS/GRAPH can also create Java output.

There are some limitations.

The following graph procedures do not produce Java output:

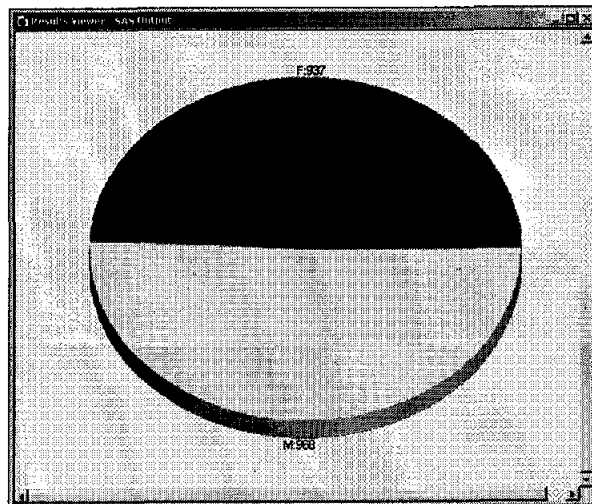
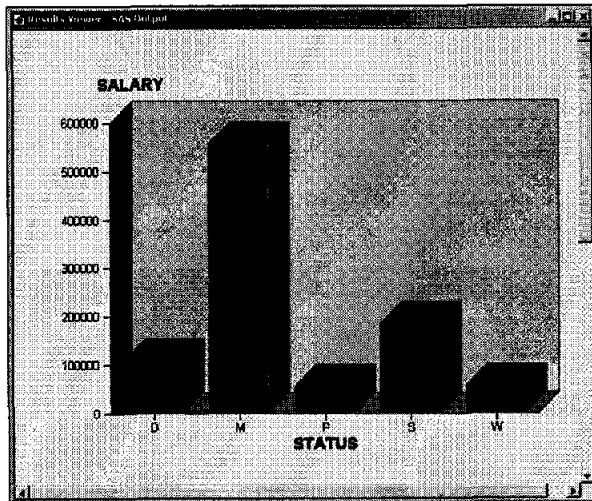
- PROC GANNO
- PROC GPRINT
- PROC GREPLAY
- PROC GSLIDE
- PROC GTESTIT

Since Version 8.1, Java output can support user-defined formats. Specific syntax is required to accommodate user-defined formats.

```

Command ==>
00001 options reset=global;
00002       reset=all;
00003       device = java transparency;
00004
00005 ods html file="c:\mygif.html";
00006 proc gchart data=saved.demog;
00007   uber3d status / sumvar = salary;
00008 run;
00009   pie3d gender / sumvar = age;
00010 run;
00011 quit;
00012 ods html close;
00013

```



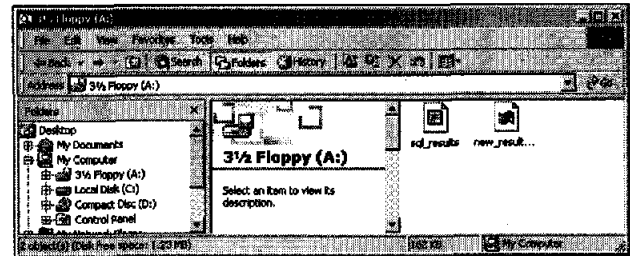
## Printing to Disk - Printer Control Language

ODS Printer has the capability to create PCL (Printer Control Language) files. This output is common to all HP printers with PCL levels 1 to 6 (level 6 is not supported in ODS Printer).

```

Command ==>
00001 ods listing close;
00002
00003 filename todisk "e:\new_results.pcl";
00004
00005 ods printer file=todisk
00006       printer=PCL5;
00007 proc sql;
00008   select *
00009   from saved.demog;
00010 quit;
00011 ods printer close;
00012
00013 ods listing;
00014

```



```

Command ==>
000
0001 ods printer file=todisk
0002       printer=PCL5;
NOTE: Writing ODS PRINTER output to file "e:\new_results.pcl" (DISK), printer
0003 proc sql;
0004   select *
0005   from saved.demog;
0006 quit;
NOTE: PROCEDURE SQL used:
      real time    9.89 seconds
      user time    0.11 seconds
0007 ods printer close;
NOTE: ODS PRINTER printed 2 pages to e:\new_results.pcl.
0008
0009 ods listing;

```

## Printing to Disk - Portable Document Files

ODS Printer can create PDF output. The process depends on the version of SAS in use, as well as converting RTF to PDF.

In SAS Version 8, generating a Postscript document using the syntax shown previously creates PDF output.

This PS document is then distilled using the fill version of Adobe Acrobat (not Adobe Reader).

In SAS Version 8e (8.1), the process to create PDF is more direct.

```

Command ==>
00001 ods listing close;
00002
00003 ods printer file="e:\8e_results.pdf" pdf;
00004 proc sql;
00005   select *
00006   from saved.demog;
00007 quit;
00008 ods printer close;
00009
00010 ods listing;
00011

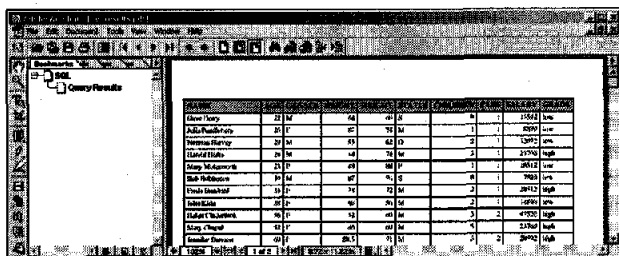
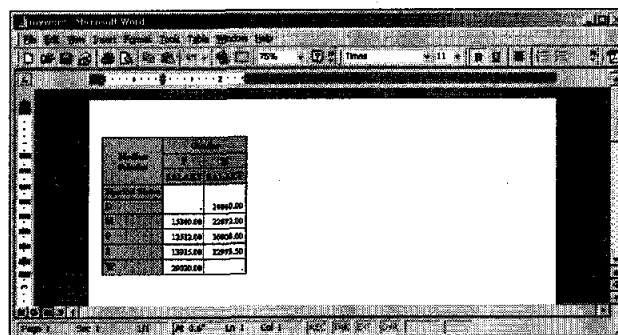
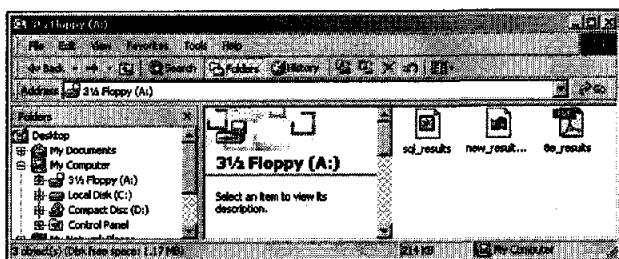
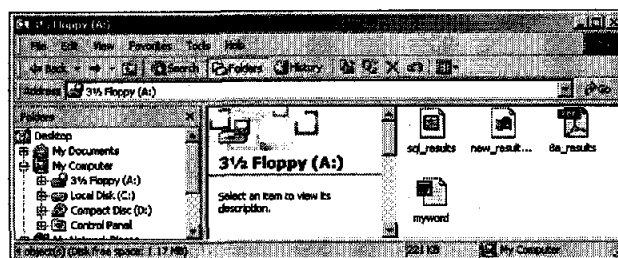
```

```

609 ods listing close;
610
611 ods printer file="a:\%e_results.pdf" pdf;
WARNING: The PDF option is experimental in this release.
NOTE: Writing ODS PRINTER output to file "a:\%e_results.pdf" [DISK], printer
612 proc sql;
613     select *
614     from saved.demog;
615 quit;
NOTE: PROCEDURE SQL used:
      real time           0.41 seconds
      cpu time            0.11 seconds

616 ods printer close;
NOTE: ODS PRINTER printed 2 pages to a:\%e_results.pdf.
617
618 ods listing;

```



## Printing to Disk - Rich Text Format

**ODS can create Rich Text Format (\*.rtf) for use in a word-processing application.**

The ODS RTF output was 'experimental' in earlier versions; Version 8e has dropped the warning.

**The results are satisfactory.**

```

Command ==>
00001 ods rtf file="a:\myword.rtf";
00002
00003 proc tabulate data=saved.demog;
00004 class status gender;
00005 var salary;
00006 table status,gender*salary*median /
00007      rts=17 box="Median Salaries";
00008 label status = "Marital Status"
00009      gender= "Gender"
00010      salary=" ";
00011 keylabel median = " ";
00012 run;
00013 ods rtf close;
00014

```

```

530 ods rtf file="e:\myword.rtf";
NOTE: Writing RTF Body file: e:\myword.rtf
533
534 proc tabulate data=saved.demog;
535 class status gender;
536 var salary;
537 table status,gender*salary*median /
538 rtr=1 box="Median Salaries";
539 label status = "Marital Status"
540 gender = "Gender"
541 salary=" ";
542 keylabel median = " ";
543 run;
NOTE: There were 50 observations read from the data set SAVED.DEMOG.
NOTE: PROCEDURE TABULATE used:
real time 0.23 seconds
cpu time 0.09 seconds

544 ods rtf close;

```