CHAPTER

3

The ODS Statements

Overview

The ODS statements control all the features of the Output Delivery System. Each ODS statement controls a different feature. You can use these statements anywhere in your SAS program. Some of them remain in effect until you explicitly change them. Others are automatically cleared at particular times (see the documentation for individual statements).

What Does Each ODS Statement Do?

To do this ...	Use this statement
Specify output objects to exclude from ODS destinations.	ODS EXCLUDE
Open, manage, or close the HTML destination. If the destination is open, you can create HTML output.	ODS HTML
Open, manage, or close the Listing destination.	ODS LISTING
Create a SAS data set from an output object and manage the selection and exclusion lists for the Output destination.	ODS OUTPUT
Specify which locations to search for definitions that were created by PROC TEMPLATE, as well as the order in which to search for them.	ODS PATH
Open, manage, or close the Printer destination. If the destination is open, you can create Printer output.	ODS PRINTER
Specify output objects for ODS destinations.	ODS SELECT
Write to the SAS log the specified selection or exclusion list.	ODS SHOW
Write to the SAS log a record of each output object that is created, or suppress the writing of this record.	ODS TRACE
Print or suppress a warning that a style definition or a table definition that is used is not supplied by SAS Institute.	ODS VERIFY

ODS EXCLUDE Statement

Specifies output objects to exclude from ODS destinations.

Default: Not using the ODS EXCLUDE statement is the same as specifying

```
ODS EXCLUDE NONE;
```

Tip: Although you can maintain a selection list for one destination and an exclusion list for another, it is easier to understand the results if you maintain the same types of lists for all the destinations that you route output to.

See also: “ODS SELECT Statement” on page 68

ODS <*ODS-destination*> **EXCLUDE** *exclusion(s)* | ALL | NONE;

Required Arguments

exclusion(s)

identifies the output objects to add to an exclusion list. By default, ODS automatically modifies exclusion lists at the end of a DATA step that uses ODS or at the end of a procedure step. For information on modifying these lists, see “Modifying Selection and Exclusion Lists” on page 25. For information ending a procedure or DATA step, see the section on DATA Step Processing in *SAS Language Reference: Concepts*.

Each *exclusion* has the following form:

output-object<(PERSIST)>

where

output-object

specifies one or more output objects to exclude.

To specify an output object, you need to know what output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. (See “ODS TRACE Statement” on page 71.) You can specify an output object as

□ a full path (shown as the path in the trace record).

□ a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For instance, if the full path is

```
Univariate.City_Pop_90.TestsForLocation
```

partial paths are

```
City_Pop_90.TestsForLocation


TestsForLocation
```

□ a label (surrounded by quotation marks).

□ a label path. (The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.)

□ a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For instance, if the label path is

```
"The UNIVARIATE Procedure"."CityPop_90"
."Tests For Location"
```

partial label paths are

```
"CityPop_90"."Tests For Location"


"Tests For Location"
```

□ a mixture of labels and paths.

□ any of these specifications followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object named TestsForLocation.

PERSIST
keeps the *output-object* that precedes it in the exclusion list, even if the DATA or procedure step ends, until you explicitly modify the list with

- □ any ODS SELECT statement
- □ ODS EXCLUDE NONE
- □ ODS EXCLUDE ALL
- □ an ODS EXCLUDE statement that applies to the same output object but does not specify PERSIST.

ALL
sets the list to EXCLUDE ALL.

Interaction: If you specify ALL without specifying a destination, ODS sets the overall list to EXCLUDE ALL and sets all other lists to their defaults.

Tip: Using EXCLUDE ALL is different from closing a destination. The destination remains open, but ODS does not send any output objects to the destination.

NONE
sets the list to EXCLUDE NONE, which has the same effect as SELECT ALL.

Interaction: If you specify NONE without specifying a destination, ODS sets the overall list to EXCLUDE NONE and sets all other lists to their defaults.

Options

ODS-destination
specifies which ODS destination's exclusion list to write to, where *ODS-destination* can be HTML, LISTING, or PRINTER (see "ODS Destinations" on page 23).

Default: If you omit *ODS-destination*, ODS writes to the overall exclusion list.

Tip: To set the exclusion list for the Output destination to something other than the default, use the ODS OUTPUT statement (see "ODS OUTPUT Statement" on page 61).

ODS HTML Statement

Opens, manages, or closes the HTML destination. If the destination is open, you can create HTML output (output that is written in Hypertext Markup Language).

Featured in: Example 2 on page 76, Example 3 on page 80, Example 4 on page 86, Example 5 on page 93, and Example 7 on page 100

ODS HTML *action*;

ODS HTML *HTML-file-specification(s) <option(s)>*;

To do this ...	Use this *action*
Close the HTML destination and any files that are associated with it	CLOSE
Select output objects for the HTML destination	SELECT

To do this ...	Use this *action*
Exclude output objects from the HTML destination	EXCLUDE
Write to the SAS log the current selection or exclusion list for the HTML destination	SHOW

To do this ...	Use this option
Specify the base name for the HTML anchor tag that identifies each output object in the current body file	ANCHOR=
Specify a string to use as the first part of all links and references that ODS creates in the HTML files	BASE=
Control the destination of the footnotes that are defined by the graphics program that generates the HTML output	GFOOTNOTE | NOGFOOTNOTE
Specify the destination for all graphics output that is generated while the HTML destination is open	GPATH=
Control the destination of the titles that are defined by the graphics program that generates the HTML output	GTITLE | NOGTITLE
Specify HTML to place between the <HEAD> and < /HEAD> tags in all the HTML files that the HTML destination writes to	HEADTEXT=
Specify HTML to use as the <META> tag inside the <HEAD> and < /HEAD> tags of all the HTML files that the HTML destination writes to	METATEXT=
Create a new body file at the specified starting-point	NEWFILE=
Specify the location (an external file or a SAS catalog) for all HTML files	PATH=
Specify an alternative record separator for the HTML files	RECORD_SEPARATOR
Specify the style definition to use in writing the HTML files	STYLE=
Translate the HTML files to the requested represention	TRANTAB=

Actions

An *action* takes some action regarding selection and exclusion lists, or closes the HTML destination. *action* can be one of the following:

CLOSE

closes the HTML destination and any files that are associated with it. When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

Featured in: Example 2 on page 76

EXCLUDE *exclusion(s)* | ALL | NONE
excludes output objects from the HTML destination.

Restriction: The HTML destination must be open for this action to take effect.

See also: "ODS EXCLUDE Statement" on page 48 and "How ODS Determines the Destinations for an Output Object" on page 25

SELECT *selection(s)* | ALL | NONE
selects output objects for the HTML destination.

Restriction: The HTML destination must be open for this action to take effect.

See also: "ODS SELECT Statement" on page 68 and "How ODS Determines the Destinations for an Output Object" on page 25

SHOW
writes to the SAS log the current selection or exclusion list for the HTML destination. If the list is the default list (SELECT ALL), SHOW also writes the current overall selection or exclusion list.

Restriction: The HTML destination must be open for this action to take effect.

Required Arguments

If you do not specify an *action*, you must use an *HTML-file-specification*.

HTML-file-specification
opens the HTML destination and specifies the HTML file or files to write to. These files remain open until you either

□ close the HTML destination with ODS HTML CLOSE

□ specify another file to write to instead.

You can specify up to four HTML files to write to. The specifications for these files have the following form:

file-type=file-specification<(file-specification-suboption(s))>

where

file-type=file-specification
associates a type of HTML file with a particular file. *file-type* can be one of the following:

BODY=
identifies the file that contains the HTML output.

Alias: FILE=

Requirement: You cannot open the HTML destination without specifying a body file. Therefore, BODY= is required if the HTML destination is closed.

Featured in: Example 2 on page 76, Example 3 on page 80, and Example 4 on page 86

CONTENTS=
identifies the file that contains a table of contents to the HTML output. The contents file links to the body file.

Featured in: Example 3 on page 80 and Example 4 on page 86

FRAME=
identifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, you see a table of contents, a table of pages, or both, as well as the body file.

Restriction: If you specify FRAME=, you must also specify CONTENTS= or PAGE= or both.

Featured in: Example 3 on page 80 and Example 4 on page 86

PAGE=

identifies the file that contains a description of each page of the body file and links to the body file. ODS produces a new page of output whenever a procedure explicitly asks for a new page. The SAS system option PAGESIZE= has no effect on pages in HTML output.

Featured in: Example 3 on page 80 and Example 4 on page 86

file-specification identifies the file or SAS catalog to write to. It can be one of the following:

'*external-file*'

is the name of an external file to write to.

Featured in: Example 3 on page 80 and Example 4 on page 86

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

Featured in: Example 7 on page 100

entry.HTML

specifies an entry in a SAS catalog to write to.

Interaction: If you specify an entry name, you must also specify a library and catalog. See the discussion of PATH= on page 57.

file-specification-suboptions

provide instructions for writing the HTML files. You specify these options inside parentheses next to the *file-specification* in the BODY=, CONTENTS=, PAGE=, or FRAME= option.

NO_BOTTOM_MATTER

By default, when you close a file that was open for HTML output, ODS writes some HTML to the bottom of the file. This HTML ends the file so that the HTML is valid and can be cleanly viewed in a browser.

If you wish to leave a body file in a state that you can append to with ODS, use NO_BOTTOM_MATTER with the *file-specification* in the BODY= option in the ODS HTML statement that opens the file. This option, in conjunction with NO_TOP_MATTER, makes it possible for you to add output to a file that already exists and to put your own HTML code in the file between output objects.

Alias: NOBOT

Interaction: When you are opening a file that ODS has previously written to, you must use ANCHOR= to specify a new base name for the anchors to avoid duplicating anchors that already exist in the file (see the discussion of ANCHOR= on page 54).

Tip: Appending to an external file requires that you use a FILENAME statement with the appropriate option for the host operating environment.

See also: NO_TOP_MATTER on page 53

Featured in: Example 7 on page 100

NO_TOP_MATTER

By default, when you open a file to create HTML output, ODS writes some HTML to the top of the file.

If ODS has already written to the body file that you are opening, you must open the file with NO_TOP_MATTER in the *file-specification* in the BODY= option in the ODS HTML statement that opens the file. This suboption, in conjunction with NO_BOTTOM_MATTER, makes it possible for you to add output to a file that already exists and to put your own HTML code in the file between output objects.

Alias: NOTOP

Interaction: When you are opening a file that ODS has previously written to, you must use ANCHOR= to specify a new base name for the anchors to avoid duplicating anchors that already exist in the file (see the discussion of ANCHOR= on page 54).

Tip: Appending to an external file requires that you use a FILENAME statement with the appropriate option for the host operating environment.

See also: NO_BOTTOM_MATTER on page 53

Featured in: Example 7 on page 100

URL='*Uniform-Resource-Locator*'

provides a URL for *file-specification*. ODS uses this URL instead of the file name in all the links and references that it creates that point to the file.

Tip: This option is useful for building HTML files that may be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), they work as long as the contents, page, and body files are all in the same location.

Tip: You never need to specify this suboption with FRAME= because no ODS file references the frame file.

See also: "HTML Links and References" on page 31

DYNAMIC

enables you to send HTML output directly to a web server instead of writing it to a file. This option sets the value of the HTMLCONTENTTYPE= attribute.

Default: If you do not specify DYNAMIC, ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction: If you specify the DYNAMIC suboption with any file specification in the ODS HTML statement, you must specify it for all the file specifications in the statement.

Interaction: *HTML-file-specification* is required if the HTML destination is closed.

Options

ANCHOR='*anchor-name*'

specifies the base name for the HTML anchor tag that identifies each output object in the current body file. Each output object must have an anchor tag for the contents, page, and frame files to link to or to reference. The links and references, which are automatically created by ODS, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR='tabulate', ODS names the first anchor **tabulate**. The second anchor is named **tabulate1**; the third is named **tabulate2**, and so on.

You can change anchor names as often as you like by submitting the ANCHOR= option in an ODS HTML statement anywhere in your program. Once you have specified an anchor name, it remains in effect until you specify a new one.

Default: IDX

Interaction: If you open an HTML file to append to it, be sure to specify a new anchor so that you don't write the same anchors to the file again. ODS cannot know about anchors that are already in a file when it opens the file.

Tip: Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your HTML output. Because you can control where the anchor name changes, you know ahead of time what the anchor name will be at those points.

See also: "How ODS Constructs Links and References" on page 34

Featured in: Example 7 on page 100

BASE='*string*'

Specifies a string to use as the first part of all links and references that ODS creates in the HTML files. Consider this specification:

```
BASE='http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string **http://www.*your-company.com/local-url*/**. The appropriate *anchor-name* completes the link.

See also: "How ODS Constructs Links and References" on page 34

Featured in: Example 5 on page 93

GFOOTNOTE | NOGFOOTNOTE

controls the destination of the footnotes that are defined by the graphics program that generates the HTML output. GFOOTNOTE includes all the currently defined footnotes within the graphics output (the GIF file) that is called by the body file. NOGFOOTNOTE suppresses all the currently defined footnotes from appearing in the GIF files. Instead, they become part of the body file.

Default: GFOOTNOTE

Restriction: Footnotes that are displayed by ODS HTML support most SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as height and text angle specifications, are ignored. For details, see "TITLE, FOOTNOTE, and NOTE Statements" in "SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference.*

Restriction: This option applies only to SAS programs that produce one or more GIF files.

GPATH=*file-specification* <(url='*Uniform-Resource-Locator*' | NONE)>

specifies the destination for all graphics output that is generated while the HTML destination is open.

file-specification

identifies the file or SAS catalog to write to. If you specify a file, it should be an aggregate storage location, such as a directory or partitioned data set. Each output object that ODS places in the file is named automatically using the SAS/GRAPH catalog entry name as the base name and incrementing the name as necessary. For more information on how ODS names catalog entries and external files, see "ODS HTML Statement" in "SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference.*

file-specification can be one of the following:

'*external-file*'

is the name of an external file to write to.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*).

Interaction: If you use a fileref in the GPATH= option, ODS does not use information from GPATH= when it constructs links. (See "How ODS Constructs Links and References" on page 34.)

libref.catalog
specifies a SAS catalog to write to.

URL='*Uniform-Resource-Locator*' | NONE
provides a URL for *file-specification*. ODS uses this URL instead of the file name in all the links and references that it creates to the file. If you specify the keyword NONE, no information from the GPATH= option appears in the links or references.

Tip: This option is useful for building HTML files that may be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), they will work as long as the contents, page, and body files are all in the same location.

See also: "How ODS Constructs Links and References" on page 34

Featured in: Example 5 on page 93

Default: If you omit GPATH=, ODS stores graphics in the location that is specified by PATH=. (See the discussion of PATH= on page 57.) If you do not specify PATH=, ODS stores the graphics in the current directory.

GTITLE | NOGTITLE
controls the destination of the titles that are defined by the graphics program that generates the HTML output. GTITLE includes all the currently defined titles within the graphics output that is called by the body file. NOGTITLE suppresses all the currently defined titles from appearing in the graphics output. Instead, they become part of the body file.

Default: GTITLE

Restriction: Titles that are displayed by ODS HTML support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as height and text angle specifications, are ignored. For details, see "TITLE, FOOTNOTE, and NOTE Statements" in "SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference.*

Restriction: This option applies only to SAS programs that produce one or more GIF files.

HEADTEXT='*HTML-for-document-head*'
specifies HTML to place between the <HEAD> and </HEAD> tags in all the HTML files that the HTML destination writes to. Use HEADTEXT= to define programs (for example, java scripts) that you can use later in the file.

Tip: ODS cannot parse the HTML that you supply. It should be well-formed HTML that makes sense in the context of the <HEAD> and </HEAD> tags. For information on HTML, refer to one of the many reference books that are available on the subject.

METATEXT='*HTML-for-document-head*'
specifies HTML to use as the <META> tag between the <HEAD> and </HEAD> tags of all the HTML files that the HTML destination writes to. This HTML provides the browser with information about the document that it is loading. This information could include things like the content-type and the character set to use. For information on HTML, refer to one of the many reference books that are available on the subject.

Default: If you do not specify METATEXT=, ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

Tip: ODS cannot parse the HTML that you supply. It should be well-formed HTML that makes sense in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, your HTML should look like this:

```
<META your-metatext-which-could-be-very-long>
```

NEWFILE=<*starting-point*>

creates a new body file at the specified *starting-point*. ODS automatically names new files by incrementing the name of the body file. For example, if you specify BODY='REPORT.HTML', ODS names the first body file REPORT.HTML. Additional body files are named REPORT1.HTML, REPORT2.HTML, and so forth. If you end the file name with a digit, ODS begins incrementing with that digit. For instance, if you specify BODY='MAY5.HTML', ODS names the first body file MAY5.HTML. Additional body files are named MAY6.HTML, MAY7.HTML, and so forth.

starting-point can be one of the following:

NONE

writes all output to the body file that is currently open.

OUTPUT

starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each GRSEG or GIF file that the program generates.

Alias: TABLE

PAGE

starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC

starts a new body file each time that you start a new procedure.

Default: NONE

Featured in: Example 5 on page 93

NOGFOOTNOTE

See the discussion of GFOOTNOTE | NOGFOOTNOTE on page 55.

NOGTITLE

See the discussion of GTITLE | NOGTITLE on page 56.

PATH=*file-specification*<(URL='*Uniform-Resource-Locator*' | NONE)>

specifies the location (an external file or a SAS catalog) for all HTML files.

file-specification

identifies the file or SAS catalog to write to. If you specify a file, it should be an aggregate storage location, such as a directory or partitioned data set.

file-specification can be one of the following:

'*external-file*'

is the name of an external file to write to.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*).

Interaction: If you use a fileref in the PATH= option, ODS does not use information from PATH= when it constructs links. (See "How ODS Constructs Links and References" on page 34.)

libname.catalog

specifies a SAS catalog to write to.

URL='*Uniform-Resource-Locator*' | NONE

provides a URL for *file-specification*. ODS uses this URL instead of the file name in all the links and references that it creates to the file. If you specify the keyword NONE, no information from the PATH= option appears in the links or references.

Tip: This option is useful for building HTML files that may be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), they work as long as the contents, page, and body files are all in the same location.

See also: “How ODS Constructs Links and References” on page 34

Featured in: Example 5 on page 93

RECORD_SEPARATOR=’*string*’ | NONE

specifies an alternative record separator for the HTML files. A record separator is the character or string that separates lines in the HTML files.

Different operating environments use different separator characters. If you don’t specify a record separator, the HTML files are formatted for the environment that you run the SAS job in. However, if you are generating files in one operating environment for viewing in another operating environment that uses a different separator character, you can specify a record separator that is appropriate for the target environment.

’*string*’

is the hexadecimal representation of one or more characters. For example, the following option specifies a record-separator of a carriage-return character and a linefeed character (on an ASCII file system):

```
RECORD_SEPARATOR='0D0A'x
```

NONE

produces HTML that is appropriate for the environment that you run the SAS job in. In many operating environments, using a value of NONE is the same as omitting the RECORD_SEPARATOR option. However, in mainframe operating environments, it is not.

Operating Environment Information: In a mainframe environment, by default, ODS produces a binary file that contains embedded record-separator characters. While this approach means that the file is not restricted by the line-length restrictions on ASCII files, it also means that if you view the file in an editor, the lines all run together.

If you want to format the HTML files so that you can read them with an editor, use RECORD_SEPARATOR=NONE. In this case, ODS writes one line of HTML at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If it isn’t, the HTML may wrap to another line at an inappropriate place. △

Alias:

RECSEP=

RS=

STYLE=’*style-definition*’

specifies the style definition to use in writing the HTML files. For more information, see “Style Definitions, Style Elements, and Style Attributes” on page 41.

Default: If you do not specify a style definition, ODS uses the file that is specified in the SAS registry subkey

ODS ▶ PREFERENCES ▶ HTML ▶ Selected Style

By default, this value specifies **styles.default**.

Featured in: Example 7 on page 100

TRANTAB=’*translation-table*’

translates the HTML files to the requested represention. This option is particularly useful if you are creating files, for example, on an EBCDIC-based operating system and if the files are destined for an ASCII-based web server. If you use TRANTAB=, you can transfer the files in binary mode because they are already in the appropriate representation. (SAS facilities for changing translation tables are much more flexible than those of FTP, for example.)

translation-table can be any translation table that SAS provides or any user-defined translation table.

See also: For information on translation tables, see the documentation of the TRANTAB= system option in the section on system options in *SAS Language Reference: Dictionary*.

For information on creating, editing, and displaying customized translation tables, see “The TRANTAB Procedure” in *SAS Procedures Guide*. You can also use PROC TRANTAB to view and modify translation tables that are supplied by SAS Institute.

ODS LISTING Statement

Opens, manages, or closes the Listing destination.

Featured in: Example 2 on page 76

ODS LISTING <*action*>;

ODS LISTING <DATAPANEL=*number* | DATA | PAGE>;

To do this ...	Use this *action*
Close the Listing destination	CLOSE
Select output objects for the Listing destination	SELECT
Exclude output objects from the Listing destination	EXCLUDE
Write to the SAS log the current selection or exclusion list for the Listing destination	SHOW

Without an Action or Options

If you use the ODS LISTING statement without an action or options, it opens the Listing destination.

Actions

An *action* takes some action regarding selection and exclusion lists, or closes the Listing destination. *action* can be one of the following:

CLOSE

closes the Listing destination. When an Output destination is closed, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

Featured in: Example 2 on page 76

EXCLUDE *exclusion(s)* | ALL | NONE

excludes output objects from the Listing destination.

Restriction: The Listing destination must be open for this action to take effect.

See also: "ODS EXCLUDE Statement" on page 48

SELECT *selection(s)* | ALL | NONE

selects output objects for the Listing destination.

Restriction: The Listing destination must be open for this action to take effect.

See also: "ODS SELECT Statement" on page 68

SHOW

writes to the SAS log the current selection or exclusion list for the Listing destination. If the list is the default list (SELECT ALL), SHOW also writes the current overall selection or exclusion list.

Restriction: The Listing destination must be open for this action to take effect.

Options

DATAPANEL=*number* | DATA | PAGE

suggests how to split a table that is too wide to fit on a single page into sections of columns and rows. Each section of columns and rows is a *data panel*. Each data panel has column headers at the top.

Note: In this context, a page is what the procedure uses as a page in creating the Listing output. The SAS system options LINESIZE= and PAGESIZE= generally determine the page size, although some procedures (PROC REPORT, for instance) can temporarily override the values that the system options specify. △

number

writes the specified number of observations in a panel, if possible. More than one panel may occur on every page if space permits.

Range: 1 to the largest integer that the operating system supports

DATA

bases the size of the panel on the way the table is stored in memory. This value provides the fastest performance. However, if the table contains many columns, the number of rows in each panel may be small.

PAGE

tries to make panels that match the page size. If the table contains more columns than can fit on a page, the first page is filled with as many observations as possible for as many columns as will fit on a single line. The second page contains the same observations for the next group of columns, and so on until all rows and columns have been printed.

This arrangement will minimize the amount of space used for column headers because most pages will contain observations for only one set of columns.

Restriction: If the page size is greater than 200, ODS uses DATAPANEL=200.

Default: PAGE

ODS OUTPUT Statement

Creates a SAS data set from an output object and manages the selection and exclusion lists for the Output destination.

Featured in: Example 3 on page 80

ODS OUTPUT *action*;

ODS OUTPUT *data-set-definition(s)*;

To do this ...	Use this *action*
Set the list for the Output destination to EXCLUDE ALL	CLEAR
Close the Output destination	CLOSE
Write to the SAS log the current selection or exclusion list for the Output destination	SHOW

Actions

An *action* takes some action regarding selection and exclusion lists, or closes the Output destination. *action* can be one of the following:

CLEAR
sets the list for the Output destination to EXCLUDE ALL.

CLOSE
closes the Output destination. When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

SHOW
writes to the SAS log the current selection or exclusion list for the Output destination. If the list is the default list (EXCLUDE ALL), SHOW also writes the current overall selection or exclusion list.

Restriction: The Output destination must be open for this action to take effect.

Required Arguments

If you do not specify an *action*, you must specify a *data-set-definition*.

data-set-definition
provides instructions for turning an output object into a SAS data set. ODS maintains a list of these definitions. (This list is the selection list for the Output destination.) For information on how ODS manages this list, see “Modifying Selection and Exclusion Lists” on page 25. Each *data-set-definition* has the following form:

output-object-specification<=SAS-data-set>

where

output-object-specification
has the following form:

output-object<(MATCH_ALL<=*macro-var-name*> PERSIST=PROC | RUN)>

where

output-object
identifies one or more output objects to turn into a SAS data set.

To specify an output object, you need to know what output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. (See "ODS TRACE Statement" on page 71.) You can specify an output object as

□ a full path (shown as the path in the trace record).

□ a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For instance, if the full path is

```
Univariate.City_Pop_90.TestsForLocation
```

partial paths are

```
City_Pop_90.TestsForLocation


TestsForLocation
```

□ a label (surrounded by quotation marks).

□ a label path. (The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.)

□ a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For instance, if the label path is

```
"The UNIVARIATE Procedure"."CityPop_90"
."Tests For Location"
```

partial label paths are

```
"CityPop_90"."Tests For Location"


"Tests For Location"
```

□ a mixture of labels and paths.

□ any of these specifications followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object named TestsForLocation.

By default, the ODS OUTPUT statement tries to put all output objects that have the same *output-path* into one SAS data set. Sometimes this process is problematic because the structure of the output object changes after the data set has been opened. For instance, some data-dependent output objects may have a different number of columns than other output objects identified by the same *output-path*. In this case, the columns that are in the first output object will determine the variables that are in the data set. Two difficulties may arise:

□ If subsequent output objects do not have a column that is in the first output object, the observations for those output objects have missing values for the variables that the ODS OUTPUT statement creates from those columns.

□ If subsequent output objects have additional columns that are not in the first output object, those columns cannot be added to the data set because the program data vector has already been written.

If the ODS OUTPUT statement has a problem putting all output objects into one data set, it returns a warning.

Tip: To create multiple data sets from the same output object, list the output object as many times as you wish. Each time that you list the output object, specify a different data set.

Featured in: Example 3 on page 80

MATCH_ALL<=*macro-var-name*>

creates a new data set for each output object. For an explanation of how ODS names these data sets, see the discussion of *SAS-data-set on page 63*.

If you specify *macro-var-name*, the system stores a list of all the data sets that it creates in the macro variable that you specify. Thus, if you want to concatenate all the data sets after the PROC step, you can use the macro variable to specify all the data sets in a DATA step.

Featured in: Example 3 on page 80

PERSIST=PROC | RUN

determines when ODS closes any data sets that it is creating and when it removes output objects from the selection list for the Output destination.

PROC

maintains the list of definitions even if the procedure or DATA step ends, until you explicitly modify it. To modify the list, use ODS OUTPUT with one or more *data-set-specifications*. To set the list for the Output destination to EXCLUDE ALL, use the following statement:

```
ods output clear;
```

Restriction: Only effective when MATCH_ALL is in effect (and you are, therefore, creating a data set for each output object).

Featured in: Example 6 on page 97

RUN

maintains the list of definitions and keeps the data sets that it is creating open even if the procedure or DATA step ends, until you explicitly modify the list.

See also: “How ODS Determines the Destinations for an Output Object” on page 25

SAS-data-set

names the output data set. You can use a one-level or two-level (with a libref) name.

If you are creating a single data set, the ODS OUTPUT statement simply uses the name that you specify. If you are creating multiple data sets with MATCH_ALL, the ODS OUTPUT statement appends digits to the name. For example, if you specify **test** as *SAS-data-set* and you create three data sets, ODS names the first data set **test**. The additional data sets are named **test1** and **test2**.

Note: If you end the file name with a digit, ODS begins incrementing with that digit. For example, if you specify **may5** as *SAS-data-set* and you create three data sets, ODS names the first data set **may5**. The additional data sets are named **may6** and **may7**. △

Default: If you do not specify a data set, ODS names the output data set DATA*n*, where *n* is the smallest integer that makes the name unique.

Tip: You can specify data set options in parentheses immediately after *SAS-data-set*.

Featured in: Example 3 on page 80

SHOW
functions just like the ODS SHOW statement except that it writes only the selection or exclusion list for the Output destination.

ODS PATH Statement

Specifies which locations to search for definitions that were created by PROC TEMPLATE, as well as the order in which to search for them.

ODS PATH *location(s)*;

Required Arguments

location(s)
specifies one or more locations to search for definitions that were created by PROC TEMPLATE. ODS searches the locations in the order that they appear on the statement. It uses the first definition that it finds that has the appropriate access mode (read, write, or update) set. Each use of the ODS PATH statement completely re-establishes the list of paths.

Each *location* has the following form:

<*libname.*>*item-store* <(READ | UPDATE | WRITE)>

<*libname*>.*item-store*
identifies an item store that contains style definitions, table definitions, or both.

(READ | UPDATE | WRITE)
specifies the access mode for the definition, where

READ
provides read-only access.

WRITE
provides write access (always creating a new template store) as well as read access.

UPDATE
provides update access (creating a new template store only if the specified one does not exist) as well as read access.

Default:

sasuser.templat(update)

sashelp.tmplmst(read)

Note: SAS stores all the style definitions that it provides in sashelp.tmplmst. △

Interaction: You can use the PATH statement in a PROC TEMPLATE step to temporarily override the ODS PATH statement (see "PATH Statement" on page 231).

Tip: If you want to be able to ignore all user-defined definitions, keep them in their own item stores so that you can leave them out of the list of items stores that ODS searches.

Featured in: Example 6 on page 265

ODS PRINTER Statement

Opens, manages, or closes the Printer destination. If the destination is open, you can create Printer output (output that is formatted for a high-resolution printer).

ODS PRINTER <*action*>;

ODS PRINTER <*option(s)*>;

To do this …	Use this *action*
Close the Printer destination and the file that is associated with it	CLOSE
Select output objects for the Printer destination	SELECT
Exclude output objects from the Printer destination	EXCLUDE
Write to the SAS log the current selection or exclusion list for the Printer destination	SHOW

To do this …	Use this option
Specify whether or not to use all the color information that the style definition provides	COLOR=
Specify the file to write to	FILE=
Specify a scaling factor to apply to all the font sizes that do not have an explicit unit of measure	FONTSCALE=
Specify that ODS use the generic postscript driver that SAS provides	POSTSCRIPT
Specify the name of the printer for which to format the Printer output.	PRINTER=
Specify that ODS use the printer drivers that SAS provides	SAS
Specify the style definition to use in writing the Printer output	STYLE=
Ensure the uniformity of column widths for all pages of Printer output	UNIFORM

Without an Action or Options

If you use the ODS PRINTER statement without an action or options, it opens the Printer destination.

Actions

An *action* takes some action regarding selection and exclusion lists, or closes the Printer destination. *action* can be one of the following:

CLOSE

closes the Printer destination and the file that is associated with it. You cannot print the file until you close the destination.

When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

Featured in: Example 2 on page 76

EXCLUDE *exclusion(s)* | ALL | NONE

excludes output objects from the Printer destination.

Restriction: The Printer destination must be open for this action to take effect.

See also: “ODS EXCLUDE Statement” on page 48

SELECT *selection(s)* | ALL | NONE

selects output objects for the Printer destination.

Restriction: The Printer destination must be open for this action to take effect.

See also: “ODS SELECT Statement” on page 68

SHOW

writes to the SAS log the current selection or exclusion list for the Printer destination. If the list is the default list (SELECT ALL), SHOW also writes the current overall selection or exclusion list.

Restriction: The Printer destination must be open for this action to take effect.

Options

COLOR=YES | NO

specifies whether or not to use all the color information that the style definition provides. In order to actually print in color, you must also

- □ use a printer that is capable of printing in color
- □ set the SAS system option COLORPRINTING.

If COLOR=NO, the Printer destination generates black and white output. It renders all text and rules in black, and it ignores all specifications for a background color from the style definition except for the purposes of determining whether to print rules for the table.

Alias: COLOR | NOCOLOR

Default: The default is determined by the value of “Use Color” in the ODS/ PRINTER subkey in the SAS registry. If the value of “Use Color” is 0, ODS generates black and white Printer output by default. If the value of “Use Color” is 1, ODS generates color Printer output by default. By default, the value of “Use Color” is 1.

Tip: If you print color output on a printer that does not support color, your output may be difficult to read.

FILE=*file-specification*

specifies the file to write to. *file-specification* can be one of the following:

'external-file'

is the name of an external file to write to.

fileref

is a fileref that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information on the FILENAME statement, see the section on statements in *SAS Language Reference: Dictionary*.)

Default: If you do not specify a file to write to, ODS writes to the file that is specified by one of two SAS system options:

- □ SYSPRINT= (if you are using the Windows operating environment and do not specify the SAS option in the ODS PRINTER statement).
- □ PRINTERPATH= (in all other cases)

If the system option does not specify a file, ODS writes to the default printer (see the discussion of PRINTER= on page 67).

Featured in: Example 2 on page 76

FONTSCALE=*percent*

specifies a scaling factor to apply to all the font sizes that do not have an explicit unit of measure. Most style definitions that SAS Institute provides specify the font size as an integer between 1 and 7. These integers are the only font-size specifications that are recognized by all the browsers that SAS supports. The Printer destination must arbitrarily select a font size for each integer. The FONTSCALE= option allows you to adjust the size as you wish.

Default: 100

POSTSCRIPT

specifies that ODS use the generic postscript driver that SAS provides.

Alias: PS

Tip: Specifying this option is equivalent to specifying both the SAS option and PRINTER=POSTSCRIPT.

PRINTER=*printer-name*

specifies the name of the printer for which to format the Printer output. *printer-name* must match a subkey in the registry. If you are using the SAS printer drivers, you can find a description of the printer, which includes its destination and device type, in

CORE ▶ PRINTING ▶ PRINTERS ▶ *printer* ▶ PRINTER SETUP ▶ OUTPUT

If you are using the Windows operating environment and you do not specify the SAS option in the ODS PRINTER statement, a description of the printer is in the Windows registry.

Note: *printer-name* is not necessarily a physical printer. It could, for instance, be a file on the disk. △

Alias: PRT

Default: If you do not specify a printer, ODS formats the Printer output for the printer that is specified by one of two SAS system options:

- □ SYSPRINT= (if you are using the Windows operating environment and do not specify the SAS option in the ODS PRINTER statement)
- □ PRINTERPATH= (in all other cases).

If the system option does not specify a printer, ODS writes to the default printer driver, as specified in the SAS registry or the Windows registry. In the SAS registry, the default printer is specified in

CORE ▶ PRINTING ▶ Default Printer

Tip: To see a list of available printers for SAS printing, use the REGEDIT command. The printers are listed in the Registry Editor window under

CORE ▶ PRINTING ▶ PRINTERS

SAS

specifies that ODS use the printer drivers that the SAS system provides.

Default: By default, when ODS is running in the Windows operating environment, it uses the printer drivers that the operating environment provides.

Restriction: This option is effective only in the Windows operating environment.

STYLE=*style-definition*

specifies the style definition to use in writing the printer output. For more information, see "Style Definitions, Style Elements, and Style Attributes" on page 41.

Default: If you do not specify a style definition, ODS uses the style definition that is specified in the SAS registry subkey

ODS ▶ PREFERENCES ▶ PRINTER ▶ Selected Style

By default, this value specifies **`styles.printer`**.

UNIFORM

ensures the uniformity of column widths for all pages of Printer output. When the UNIFORM option is in effect, ODS reads the entire table before it starts to print it so that it can determine the column widths that are necessary to accommodate all the data. It uses these column widths on all pages.

Default: If you do not specify UNIFORM, ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while processing very large tables. However, it can also mean that column widths vary from one page to the next.

Tip: The UNIFORM option may cause SAS to run out of memory if you are printing a very large table. If this happens, you can explicitly set the width of each of the columns in the table and let the table print one page at a time. To do so, you must edit the table definition that you use (see "EDIT Statement" on page 227).

ODS SELECT Statement

Specifies output objects for ODS destinations.

Default: Not using the ODS SELECT statement is the same as specifying

```
ODS SELECT ALL;
```

Tip: Although you can maintain a selection list for one destination and an exclusion list for another, it is easier to understand the results if you maintain the same types of lists for all the destinations that you route output to.

See also: "ODS EXCLUDE Statement" on page 48

Featured in: Example 2 on page 76 and Example 4 on page 86

ODS SELECT <*ODS-destination*> *selection(s)* | ALL | NONE;

Required Arguments

selection(s)

identifies output objects to add to a selection list. ODS sends the items in the selection list to all active ODS destinations. By default, ODS automatically modifies selection lists when a DATA step that uses ODS or a procedure step ends. For information on modifying these lists, see "Modifying Selection and Exclusion Lists" on page 25. For information ending DATA and procedure steps, see the section on DATA Step Processing in *SAS Language Reference: Concepts*.

Each *selection* has the following form:

output-object <(PERSIST)>

where

output-object

specifies one or more output objects to select.

To specify an output object, you need to know what output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. (See "ODS TRACE Statement" on page 71.) You can specify an output object as

□ a full path (shown as the path in the trace record).

□ a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For instance, if the full path is

```
Univariate.City_Pop_90.TestsForLocation
```

partial paths are

```
City_Pop_90.TestsForLocation


TestsForLocation
```

□ a label (surrounded by quotation marks).

□ a label path. (The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.)

□ a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For instance, if the label path is

```
"The UNIVARIATE Procedure"."CityPop_90"
."Tests For Location"
```

partial label paths are

```
"CityPop_90"."Tests For Location"


"Tests For Location"
```

□ a mixture of labels and paths.

□ any of these specifications followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object named TestsForLocation.

PERSIST
keeps the *output-object* that precedes it in the selection list, even if the DATA or procedure step ends, until you explicitly modify the list with

- □ any ODS EXCLUDE statement
- □ ODS SELECT NONE
- □ ODS SELECT ALL
- □ an ODS SELECT statement that applies to the same output object but does not specify PERSIST.

Featured in: Example 4 on page 86

ALL
sets the list to SELECT ALL.

Interaction: If you specify ALL without specifying a destination, ODS sets the overall list to SELECT ALL and sets all other lists to their defaults.

Featured in: Example 4 on page 86

NONE
sets the list to SELECT NONE.

Interaction: If you specify NONE without specifying a destination, ODS sets the overall list to SELECT NONE and sets all other lists to their defaults.

Tip: Using NONE is different from closing a destination. The Output destination is still running, but ODS is restricting the output that it sends to the destination.

Options

ODS-destination
specifies which ODS destination's selection list to write to, where *ODS-destination* can be HTML, LISTING, or PRINTER. (See “ODS Destinations” on page 23.)

Default: If you omit *ODS-destination*, ODS writes to the overall selection list.

Tip: To set the selection list for the Output destination to something other than the default, use the ODS OUTPUT statement (see “ODS OUTPUT Statement” on page 61).

Featured in: Example 2 on page 76

ODS SHOW Statement

Writes to the SAS log the specified selection or exclusion list.

Featured in: Example 4 on page 86

ODS <*ODS-destination*> **SHOW**;

Options

ODS-destination

specifies which ODS destination's selection or exclusion list to write to the SAS log, where *ODS-destination* can be HTML, LISTING, OUTPUT, or PRINTER. (See "ODS Destinations" on page 23.)

Default: If you omit *ODS-destination*, ODS SHOW writes the overall selection or exclusion list.

ODS TRACE Statement

Writes to the SAS log a record of each output object that is created, or suppresses the writing of this record.

Default: OFF

Featured in: Example 1 on page 73

ODS TRACE ON< / <*option(s)*>>;

ODS TRACE OFF;

Required Arguments

OFF

turns off the writing of the trace record.

Featured in: Example 1 on page 73

ON

turns on the writing of the trace record.

Alias: OUTPUT

Featured in: Example 1 on page 73

Options

LABEL

includes the label path for the output object in the record. You can use a label path anywhere that you can use a path.

Tip: This option is most useful for users who are running a localized version of SAS because the labels are translated from English to the local language. The names and paths of output objects are not translated because they are part of the syntax of the Output Delivery System.

Featured in: Example 1 on page 73

LISTING

writes the trace record to the Listing destination, so that each part of the trace record immediately precedes the output object that it describes.

Contents of the Trace Record

ODS produces an output object by combining data (the data component) with a table definition. The trace record provides information about the data component, the table definition, and the output object. By default, the record that the ODS TRACE statement produces contains these items:

Name
: is the name of the output object. You can use the name to reference this output object and others with the same name. (For details on how to reference an output object, see “How to Specify an Output Object” on page 24.) For example, you could use this name in an ODS OUTPUT statement to make a data set from the output object, or you could use it in an ODS SELECT or an ODS EXCLUDE statement.

  Tip: The name is the rightmost part of the path that appears in the trace record.

Label
: briefly describes the contents of the output object. This label also identifies the output object in the Results folder in the Results window.

Data name
: is the name of the data component that was used to create this output object. The data name appears only if it differs from the name of the output object.

Data label
: describes the contents of the data.

Template
: is the name of the table definition that ODS used to format the output object. You can modify this definition with PROC TEMPLATE (see “EDIT Statement” on page 227).

Path
: is the path of the output object. You can use the path to reference this output object. For example, you could use the path in the ODS OUTPUT statement to make a data set from the output, or you could use it in an ODS SELECT or an ODS EXCLUDE statement.

The LABEL option modifies the trace record by including the label path for the object in the record. See the discussion of the LABEL on page 71 option.

ODS VERIFY Statement

Prints or suppresses a warning that a style definition or a table definition that is used is not supplied by SAS Institute.

ODS VERIFY <ON | OFF | ERROR>;

OPTIONS

ON | OFF | ERROR

ON
: prints the warning and sends output objects to open destinations.

OFF
: suppresses the warning.

ERROR
prints the warning and does not send output objects to open destinations.

Default: If you do not specify the ODS VERIFY statement, ODS runs with the verification process turned off. If you specify the ODS VERIFY statement but do not specify an argument, ODS runs with verification turned on.

Tip: For information on how to ignore user-created definitions, see “ODS PATH Statement” on page 64.

Why Use the ODS VERIFY Statement?

PROC TEMPLATE can modify the values in an output object. None of the definitions that SAS Institute provides modifies any values. If you receive a warning from the ODS VERIFY statement, you might want to look at the source code to verify that the values have not been modified.

Examples

Example 1: Determining What Output Objects a Procedure Creates

ODS features:

ODS TRACE statement:

LABEL
OFF
ON

Other SAS features: PROC UNIVARIATE

This example shows how to determine the names and labels of the output objects that a procedure creates. You can use this information to select and exclude output objects (see Example 2 on page 76) and to create output data sets (see Example 3 on page 80).

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, “Alternative ODS HTML Statements for Running Examples in Different Operating Environments,” on page 275. △

Program

The data set STATEPOP contains information from the 1980 and 1990 U.S. Census on the population in metropolitan and nonmetropolitan areas. The 50 states and District of Columbia are divided into four geographic regions. The data are organized by state within each region. The metropolitan and nonmetropolitan population counts are stored in one observation for both census years. A DATA step on page 282 creates this data set.

```
data statepop;
   input State $ CityPop_80 CityPop_90
         NonCityPop_80 NonCityPop_90 Region;
   format region 1.;
   label citypop_80=   ’1980 metropolitan pop in millions’
         noncitypop_80=’1980 nonmetropolitan pop in millions’
         citypop_90=   ’1990 metropolitan pop in millions’
         noncitypop_90=’1990 nonmetropolitan pop in million’
         region=’Geographic region’;
  datalines;
ME    .405    .443   .721   .785  1
NH    .535    .659   .386   .450  1
VT    .133    .152   .378   .411  1

   … more lines of data …

CA  22.907  28.799   .760   .961  4
AK    .174    .226   .227   .324  4
HI    .763    .836   .202   .272  4
;
```

This ODS TRACE statement writes the trace record to the SAS log. The LABEL option includes label paths in the trace record.

```
ods trace on / label;
```

PROC UNIVARIATE computes the univariate statistics for two variables, CityPop_80 and CityPop_90. As PROC UNIVARIATE sends each output object to the Output Delivery System, ODS writes the pertinent information for that output object to the trace record.

```
proc univariate data=statepop mu0=3.5;
   var citypop_90 citypop_80;
run;
```

This ODS TRACE statement stops the writing of the trace record to the SAS log.

```
ods trace off;
```

SAS Log

This partial SAS log shows the trace record that the ODS TRACE statement creates. PROC UNIVARIATE creates five output objects for each analysis variable: **`Moments`**, **`BasicMeasures`**, **`TestsForLocation`**, **`Quantiles`**, and **`ExtremeObs`**.

Notice that an output object has the same name and label, regardless of which variable is analyzed. You can, therefore, select all the moments tables that PROC UNIVARIATE produces by using the name or label in an ODS SELECT statement. On the other hand, the path and label path are unique for each output object because they include the name of the variable that is analyzed. You can, therefore, select an individual moments table by using the path or the label path in an ODS SELECT statement.

```
Output Added:
-------------
Name:       Moments
Label:      Moments
Template:   base.univariate.Moments
Path:       Univariate.CityPop_90.Moments
Label Path: "The Univariate Procedure"."CityPop_90"."Moments"
-------------

Output Added:
-------------
Name:       BasicMeasures
Label:      Basic Measures of Location and Variability
Template:   base.univariate.Measures
Path:       Univariate.CityPop_90.BasicMeasures
Label Path: "The Univariate Procedure"."CityPop_90"."Basic Measures of Location and Variability"
-------------

Output Added:
-------------
Name:       TestsForLocation
Label:      Tests For Location
Template:   base.univariate.Location
Path:       Univariate.CityPop_90.TestsForLocation
Label Path: "The Univariate Procedure"."CityPop_90"."Tests For Location"
-------------

Output Added:
-------------
Name:       Quantiles
Label:      Quantiles
Template:   base.univariate.Quantiles
Path:       Univariate.CityPop_90.Quantiles
Label Path: "The Univariate Procedure"."CityPop_90"."Quantiles"
-------------

Output Added:
-------------
Name:       ExtremeObs
Label:      Extreme Observations
Template:   base.univariate.ExtObs
Path:       Univariate.CityPop_90.ExtremeObs
Label Path: "The Univariate Procedure"."CityPop_90"."Extreme Observations"
-------------
```

```
Output Added:
-------------
Name:       Moments
Label:      Moments
Template:   base.univariate.Moments
Path:       Univariate.CityPop_80.Moments
Label Path: "The Univariate Procedure"."CityPop_80"."Moments"
-------------

Output Added:
-------------
Name:       BasicMeasures
Label:      Basic Measures of Location and Variability
Template:   base.univariate.Measures
Path:       Univariate.CityPop_80.BasicMeasures
Label Path: "The Univariate Procedure"."CityPop_80"."Basic Measures of Location and Variability"
-------------

Output Added:
-------------
Name:       TestsForLocation
Label:      Tests For Location
Template:   base.univariate.Location
Path:       Univariate.CityPop_80.TestsForLocation
Label Path: "The Univariate Procedure"."CityPop_80"."Tests For Location"
-------------

Output Added:
-------------
Name:       Quantiles
Label:      Quantiles
Template:   base.univariate.Quantiles
Path:       Univariate.CityPop_80.Quantiles
Label Path: "The Univariate Procedure"."CityPop_80"."Quantiles"
-------------

Output Added:
-------------
Name:       ExtremeObs
Label:      Extreme Observations
Template:   base.univariate.ExtObs
Path:       Univariate.CityPop_80.ExtremeObs
Label Path: "The Univariate Procedure"."CityPop_80"."Extreme Observations"
-------------
```

Example 2: Selecting Output for the HTML and Printer Destinations

ODS features:

- ODS HTML statement:
  - BODY=
  - CLOSE
- ODS PRINTER statement:
  - FILE=
  - CLOSE
- ODS LISTING statement:
  - CLOSE
- ODS SELECT statement:
  - with path
  - with label
  - PERSIST

Other SAS features: PROC UNIVARIATE

Data set: STATEPOP on page 73

This example selects three output objects to send to both the HTML destination and the Printer destination.

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, “Alternative ODS HTML Statements for Running Examples in Different Operating Environments,” on page 275. △

Program

The ODS LISTING statement closes the Listing destination to conserve resources. (By default, the Listing destination is open.)

```
ods listing close;
```

The OPTIONS statement controls several aspects of the Printer output. None of these options affects the HTML output.

```
options nodate nonumber;
```

The ODS HTML statement opens the HTML destination and creates HTML output. It sends all output objects to the external file **odsselect-body.htm** in the current directory. Some browsers require an extension of HTM or HTML on the filename.

```
ods html body='odsselect-body.htm';
```

The ODS PRINTER statement opens the Printer destination and creates Printer output. It sends all output objects to the external file **odsselect.ps** in the current directory.

```
 ods printer file='odsselect.ps';
```

The ODS SELECT statement specifies three output objects to send to all open destinations. The first output object is selected by its name, **BasicMeasures**. The second is selected by its label, **Tests For Location**. These two selection criteria select the output objects for the analysis of both variables. The third output object is selected by its full path **Univariate.CityPop_90.ExtremeObs**. This selection criterion selects the output object for only one variable, **CityPop_90**.

```
ods select BasicMeasures
           "Tests For Location"
           Univariate.CityPop_90.ExtremeObs;
```

PROC UNIVARIATE computes the univariate statistics for two variables, CityPop_80 and CityPop_90. As PROC UNIVARIATE sends each output object to the Output Delivery System, ODS routes the selected output objects to the HTML and Printer destinations.

```
proc univariate data=statepop mu0=3.5;
   var citypop_90 citypop_80;
run;
```

The ODS HTML statement closes the HTML destination and all the files that are associated with it. You must close the destination before you can view the output with a browser. The ODS PRINTER statement closes the Printer destination. You must close the destination before you can print the output on a physical printer.

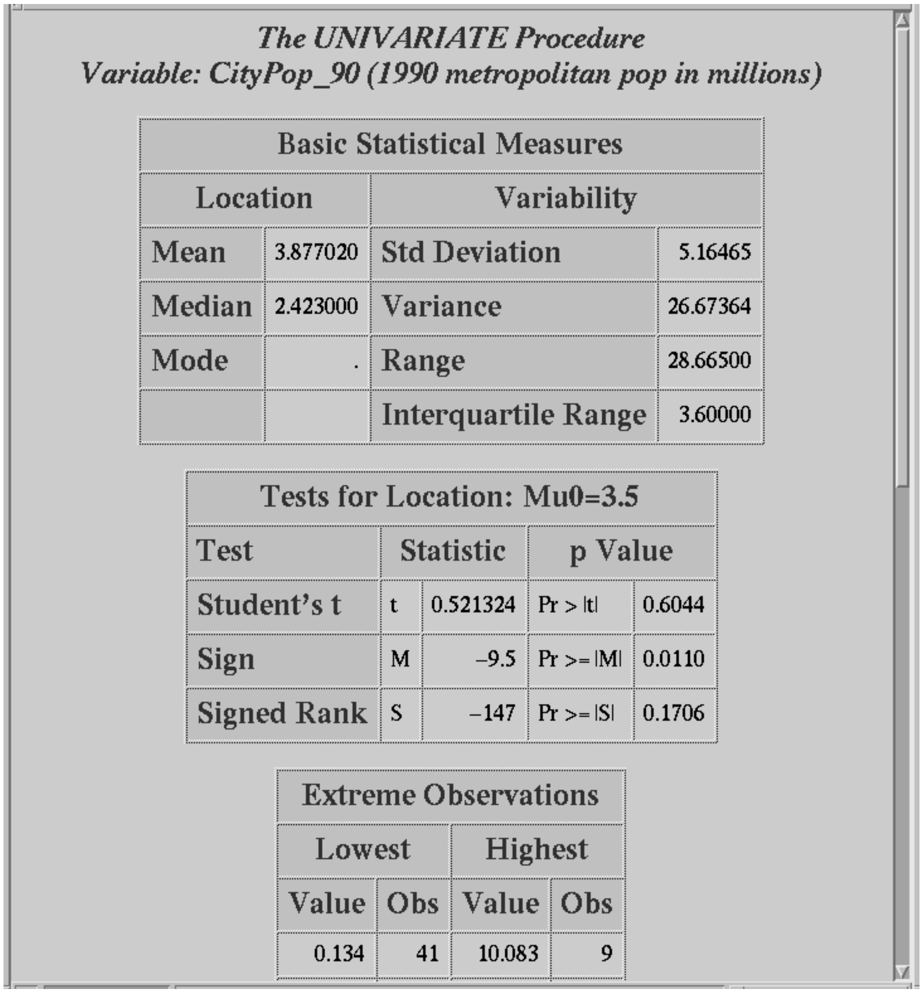
```
ods html close;

ods printer close;
```

The ODS LISTING statement opens the Listing destination to return ODS to its default setup.

```
ods listing;
```

HTML Output

The HTML output includes three output objects for the variable CityPop_90 and two output objects for the variable CityPop_80. Only the objects for **`CityPop_90`** are shown here.



The UNIVARIATE Procedure
Variable: CityPop_90 (1990 metropolitan pop in millions)

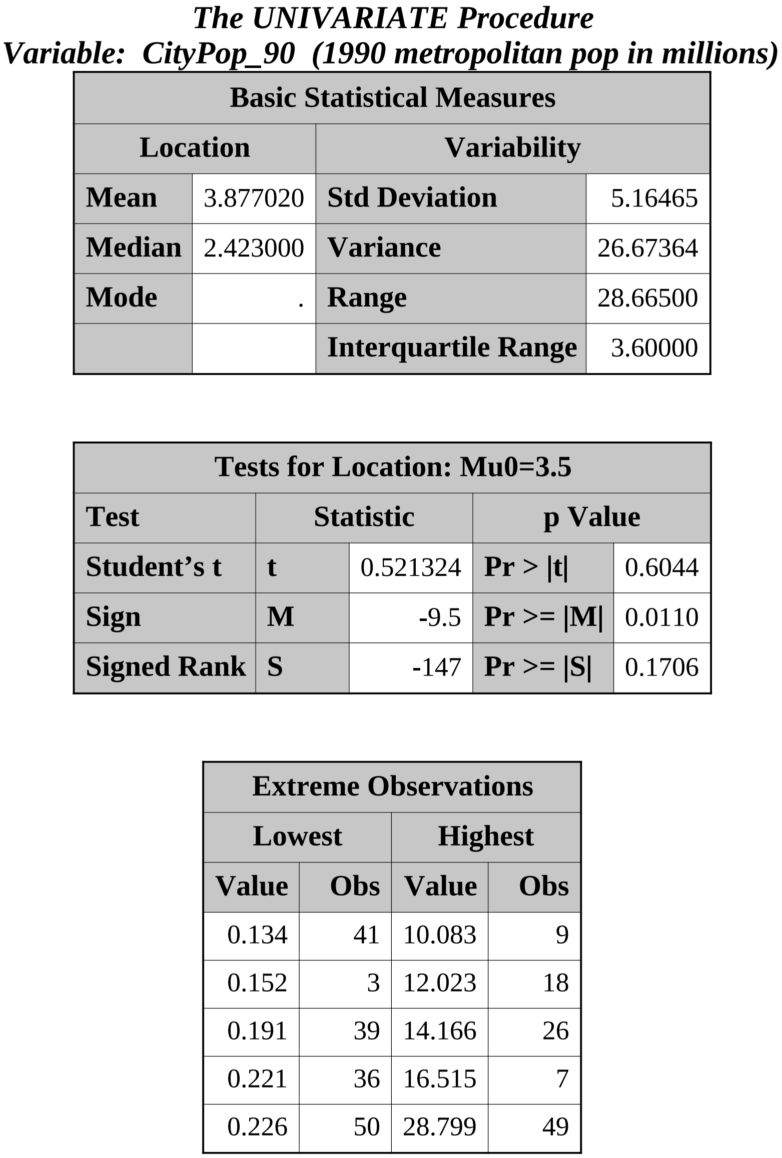
Basic Statistical Measures			
Location		Variability	
Mean	3.877020	Std Deviation	5.16465
Median	2.423000	Variance	26.67364
Mode	.	Range	28.66500
		Interquartile Range	3.60000

Tests for Location: Mu0=3.5				
Test	Statistic		p Value	
Student's t	t	0.521324	Pr > |t|	0.6044
Sign	M	-9.5	Pr >= |M|	0.0110
Signed Rank	S	-147	Pr >= |S|	0.1706

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0.134	41	10.083	9

Printer Output

The Printer output includes three output objects for the variable CityPop_90 and two output objects for the variable CityPop_80.



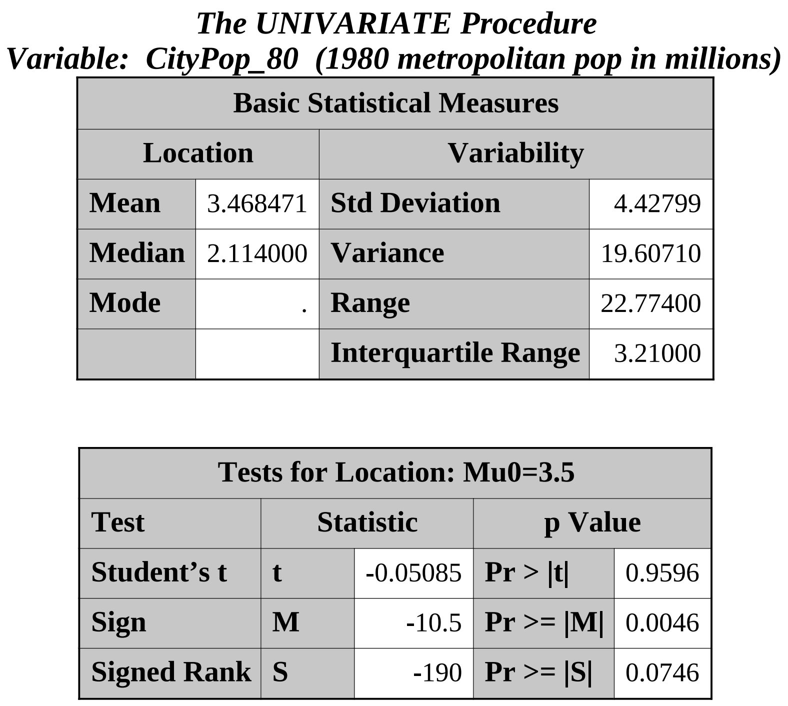
The UNIVARIATE Procedure

Variable: CityPop_90 (1990 metropolitan pop in millions)

Basic Statistical Measures			
Location		**Variability**	
Mean	3.877020	**Std Deviation**	5.16465
Median	2.423000	**Variance**	26.67364
Mode	.	**Range**	28.66500
		Interquartile Range	3.60000

Tests for Location: Mu0=3.5				
Test	**Statistic**		**p Value**	
Student's t	**t**	0.521324	**Pr > |t|**	0.6044
Sign	**M**	-9.5	**Pr >= |M|**	0.0110
Signed Rank	**S**	-147	**Pr >= |S|**	0.1706

Extreme Observations			
Lowest		**Highest**	
Value	**Obs**	**Value**	**Obs**
0.134	41	10.083	9
0.152	3	12.023	18
0.191	39	14.166	26
0.221	36	16.515	7
0.226	50	28.799	49



The UNIVARIATE Procedure
Variable: CityPop_80 (1980 metropolitan pop in millions)

Basic Statistical Measures			
Location		**Variability**	
Mean	3.468471	**Std Deviation**	4.42799
Median	2.114000	**Variance**	19.60710
Mode	.	**Range**	22.77400
		Interquartile Range	3.21000

Tests for Location: Mu0=3.5				
Test	**Statistic**		**p Value**	
Student's t	**t**	-0.05085	**Pr > |t|**	0.9596
Sign	**M**	-10.5	**Pr >= |M|**	0.0046
Signed Rank	**S**	-190	**Pr >= |S|**	0.0746

Example 3: Creating Output Data Sets

ODS features:

ODS HTML statement:

BODY=
CONTENTS=
FRAME=
PAGE=

ODS LISTING statement:

CLOSE

ODS OUTPUT statement:

data set options for the output data sets
match_all=
storing the names of the output data sets in a macro variable

Other SAS features:

PROC FORMAT

PROC PRINT

SET statement

KEEP= data set option

RENAME= data set option

This example routes two output objects that PROC TABULATE produces to both the Output destination and the HTML destination. The result is two output data sets and two output objects formatted as HTML output. PROC PRINT prints the output data sets, and the resulting output object is also routed to the HTML destination. Finally, a

DATA step combines both the output data sets into one data set, and a PROC PRINT step prints this data set as well.

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, "Alternative ODS HTML Statements for Running Examples in Different Operating Environments," on page 275. △

Program

The data set ENERGY contains data on expenditures on energy for business and residential customers in individual states in the Northeast and West regions of the United States.

```
data energy;
   length State $2;
   input Region Division state $ Type Expenditures @@;
   datalines;
1 1 ME 1 708  1 1 ME 2 379  1 1 NH 1 597   1 1 NH 2 301
1 1 VT 1 353  1 1 VT 2 188  1 1 MA 1 3264  1 1 MA 2 2498
1 1 RI 1 531  1 1 RI 2 358  1 1 CT 1 2024  1 1 CT 2 1405
1 2 NY 1 8786 1 2 NY 2 7825 1 2 NJ 1 4115  1 2 NJ 2 3558
1 2 PA 1 6478 1 2 PA 2 3695 4 3 MT 1 322   4 3 MT 2 232
4 3 ID 1 392  4 3 ID 2 298  4 3 WY 1 194   4 3 WY 2 184
4 3 CO 1 1215 4 3 CO 2 1173 4 3 NM 1 545   4 3 NM 2 578
4 3 AZ 1 1694 4 3 AZ 2 1448 4 3 UT 1 621   4 3 UT 2 438
4 3 NV 1 493  4 3 NV 2 378  4 4 WA 1 1680  4 4 WA 2 1122
4 4 OR 1 1014 4 4 OR 2 756  4 4 CA 1 10643 4 4 CA 2 10114
4 4 AK 1 349  4 4 AK 2 329  4 4 HI 1 273   4 4 HI 2 298
;
```

PROC FORMAT creates formats for Region, Division, and Type.

```
proc format;
   value regfmt 1='Northeast'
                2='South'
                3='Midwest'
                4='West';
   value divfmt 1='New England'
                2='Middle Atlantic'
                3='Mountain'
                4='Pacific';
   value usetype 1='Residential Customers'
                 2='Business Customers';
run;
```

The ODS LISTING statement closes the Listing destination to conserve resources. The Listing destination is open by default.

```
ods listing close;
```

The ODS OUTPUT statement creates SAS data sets from the output objects that PROC TABULATE produces. The name of each output object is **Report**. The MATCH_ALL option produces a SAS data set for each instance of **Report**. The name of the first data set is **energyoutput**, and the name of the second data set is **energyoutput1**. ODS stores a list of these names in the macro variable **datasetnames**. This variable is used later in the example to combine the data sets.

The KEEP= data set option limits the variables in the output data set to Region, Division, Type, and Expenditures_sum. The variable name Expenditures_sum is generated by PROC TABULATE to indicate that the variable Expenditures was used to calculate the **sum** statistic. The RENAME= data set option changes the name of the variable to Expenditures. (To learn the names of the variables in an output data set that ODS creates, use PROC CONTENTS or the Explorer window.)

```
ods output Report(match_all=datasetnames)
           =energyoutput(keep=region division type expenditures_sum
                         rename=(expenditures_sum=Expenditures));
```

The ODS HTML statement opens the HTML destination and creates HTML output. The output from PROC TABULATE goes to the body file. FRAME=, CONTENTS=, and PAGE= create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame.

```
ods html body=’odsoutput-body.htm’
        frame=’odsoutput-frame.htm’
     contents=’odsoutput-contents.htm’
         page=’odsoutput-page.htm’;
```

This PROC TABULATE step produces two output objects: one for each BY group. Because the HTML destination is open, ODS writes the output to the body file. At the same time, because the Output destination is open and because a data set name is associated with each output object, ODS creates an output data set from each BY group.

```
proc tabulate data=energy format=dollar12.;
   by region;
   class division type;
   var expenditures;
   table division,
         type*expenditures;

   format region regfmt. division divfmt. type usetype.;
   title ’Energy Expenditures for Each Region’;
   title2 ’(millions of dollars)’;
run;
```

The ODS HTML statement closes the original body file and opens a new one. The contents, page, and frame files remain open. The contents and page files will contain links to both body files.

```
ods html body=’odsoutput-printbody.htm’;
```

The PROC PRINT steps print the two output data sets that ODS created. The output from these two steps is written to the current body file, **odsoutput-printbody.htm**.

```
proc print data=energyoutput noobs;
   title ’Output Data Set from the First BY Group’;
run;
```

```
proc print data=energyoutput1 noobs;
   title 'Output Data Set from the Second BY Group';
run;
```

This DATA step combines all the data sets that were created by the ODS OUTPUT statement. (The macro variable DATASETNAMES contains the list of names.)

```
data new;
   set &datasetnames;
run;
```

This PROC PRINT step prints the data set NEW. The output is added to the current body file, **`odsoutput-printbody.htm`**.

```
proc print data=new noobs;
   title 'Combined Output Data Sets';
run;
```

This ODS HTML statement closes the HTML destination and all the files that are associated with it. To return ODS to its default setup, the ODS OUTPUT statement closes the Output destination, and the ODS LISTING statement opens the Listing destination.

```
ods html close;
ods output close;
ods listing;
```

HTML Output

Display 3.1 Initial View of the Frame File

Links to each output object appear in the table of contents and the table of pages on the left-hand side of the frame. The HTML that PROC TABULATE produced is partially visible here. Clicking on the links in the table of contents or table of pages displays the HTML output that was produced by the other procedures in this example.

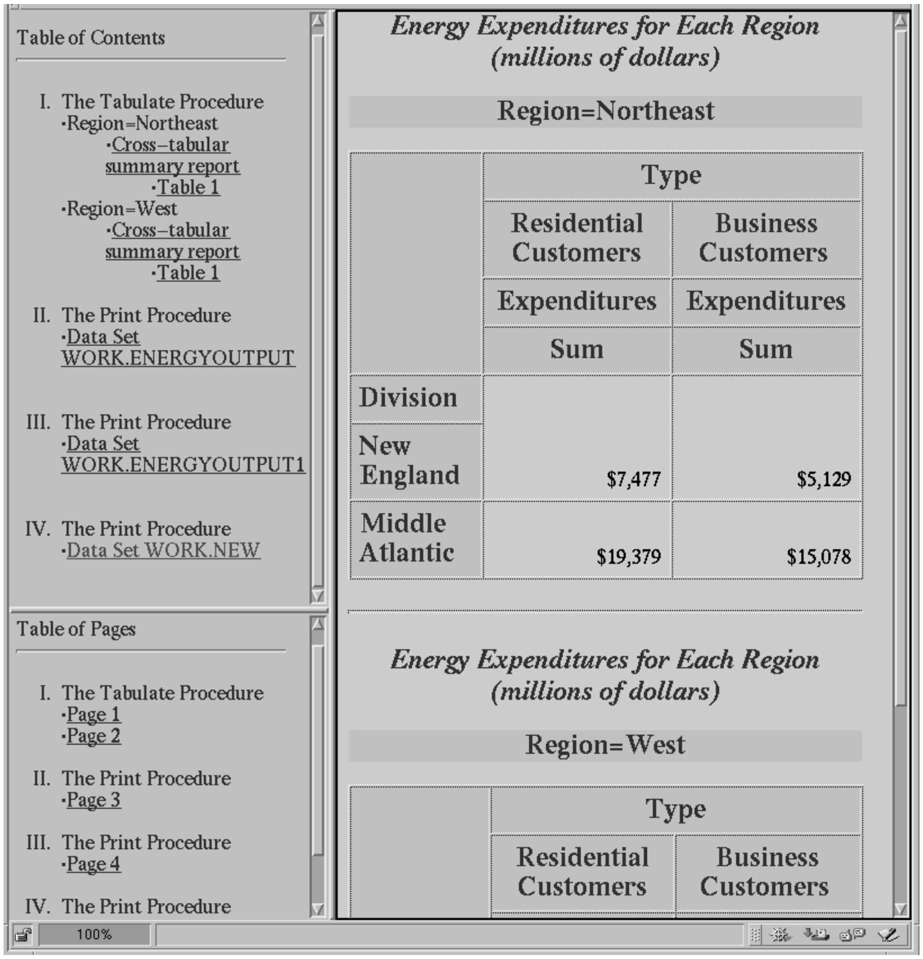


Table of Contents

I. The Tabulate Procedure
- Region=Northeast
  - Cross–tabular summary report
    - Table 1
- Region=West
  - Cross–tabular summary report
    - Table 1

II. The Print Procedure
- Data Set WORK.ENERGYOUTPUT

III. The Print Procedure
- Data Set WORK.ENERGYOUTPUT1

IV. The Print Procedure
- Data Set WORK.NEW

Table of Pages

I. The Tabulate Procedure
- Page 1
- Page 2

II. The Print Procedure
- Page 3

III. The Print Procedure
- Page 4

IV. The Print Procedure

Energy Expenditures for Each Region
(millions of dollars)

Region=Northeast

	Type	
	Residential Customers	Business Customers
	Expenditures	Expenditures
	Sum	Sum
Division		
New England	$7,477	$5,129
Middle Atlantic	$19,379	$15,078

Energy Expenditures for Each Region
(millions of dollars)

Region=West

	Type	
	Residential Customers	Business Customers

Display 3.2 Frame File After Selecting a Link

After you select the link to the data set WORK.ENERGYOUTPUT, the HTML output that PROC PRINT produced appears at the top of the right-hand side of the frame.

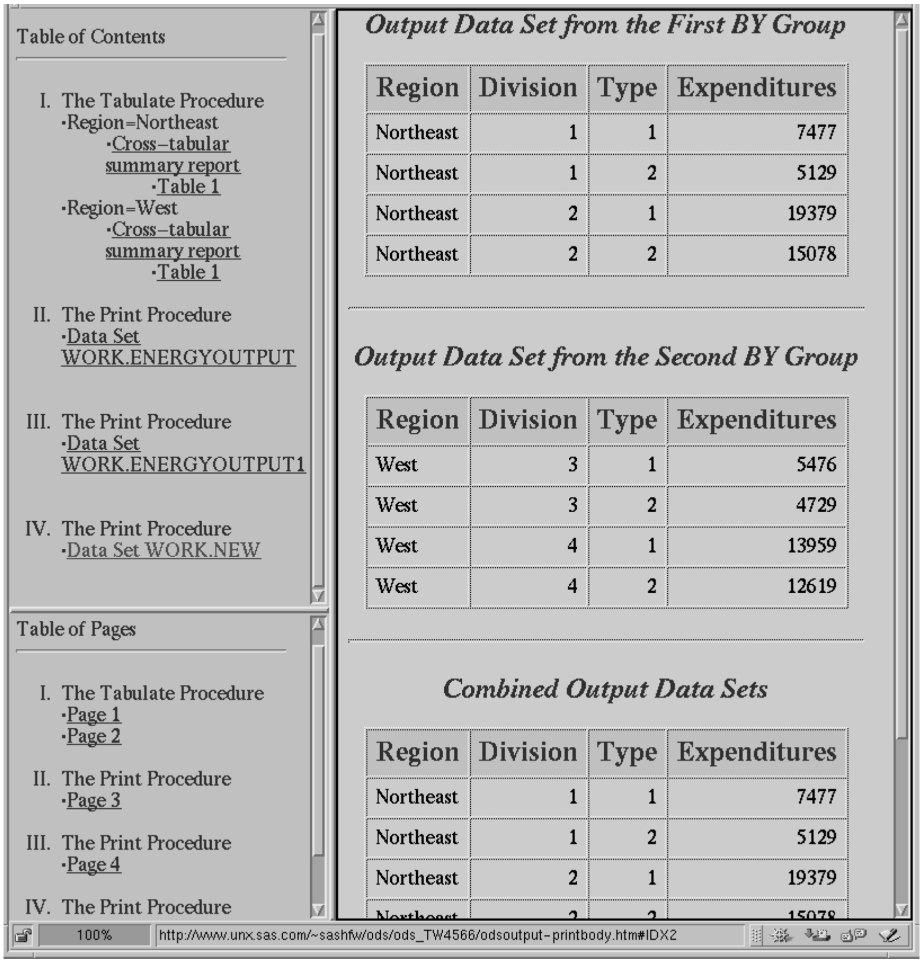


Table of Contents

I. The Tabulate Procedure
- Region=Northeast
  - Cross–tabular summary report
    - Table 1
- Region=West
  - Cross–tabular summary report
    - Table 1

II. The Print Procedure
- Data Set WORK.ENERGYOUTPUT

III. The Print Procedure
- Data Set WORK.ENERGYOUTPUT1

IV. The Print Procedure
- Data Set WORK.NEW

Table of Pages

I. The Tabulate Procedure
- Page 1
- Page 2

II. The Print Procedure
- Page 3

III. The Print Procedure
- Page 4

IV. The Print Procedure

Output Data Set from the First BY Group

Region	Division	Type	Expenditures
Northeast	1	1	7477
Northeast	1	2	5129
Northeast	2	1	19379
Northeast	2	2	15078

Output Data Set from the Second BY Group

Region	Division	Type	Expenditures
West	3	1	5476
West	3	2	4729
West	4	1	13959
West	4	2	12619

Combined Output Data Sets

Region	Division	Type	Expenditures
Northeast	1	1	7477
Northeast	1	2	5129
Northeast	2	1	19379
Northeast	2	2	15078

http://www.unx.sas.com/~sashfw/ods/ods_TW4566/odsoutput-printbody.htm#IDX2

Example 4: Using a Selection List with Multiple Procedure Steps

ODS features:

- ODS SELECT statement:
  - with label
  - with name
  - with and without PERSIST
  - ALL
- ODS SHOW statement
- ODS HTML statement:
  - BODY=
  - CONTENTS=
  - FRAME=
  - PAGE=

Other SAS features:

- PROC GLM
- PROC PRINT
- PROC PLOT

This example runs the same procedures multiple times to illustrate how ODS maintains and modifies a selection list. The ODS SHOW statement writes the overall selection list to the SAS log. The example does not alter selection lists for individual destinations, so the contents file that is generated by the ODS HTML statement shows which output objects are routed to both the HTML and the Listing destinations. (The Printer destination is not open.)

Note: This example uses PROC GLM, which is part of SAS/STAT software. △

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, "Alternative ODS HTML Statements for Running Examples in Different Operating Environments," on page 275. △

Program

The data set IRON contains data from Draper and Smith (1998, p. 98).[*] Thirteen specimens of 90/10 copper-nickel alloys were tested in a corrosion-wheel setup in order to examine corrosion. Each specimen had a certain iron content. The wheel was rotated in salt sea water at 30 ft/sec for 60 days. Weight loss was used to quantify the corrosion. FE represents the iron content, and LOSS denotes the weight loss in milligrams/square decimeter/day.

```
data iron;
   input Fe Loss @@;
   datalines;
```

[*] Draper, N. and Smith, H. (1998), *Applied Regression Analysis, Second Edition*, New York: John Wiley & Sons.

```
0.01 127.6   0.48 124.0   0.71 110.8   0.95 103.9
1.19 101.5   0.01 130.1   0.48 122.0   1.44  92.3
0.71 113.1   1.96  83.7   0.01 128.0   1.44  91.4
1.96  86.2
;
```

The ODS HTML statement opens the HTML destination and creates HTML output. The output from the procedures goes to the file **odspersist-body.htm**. FRAME=, CONTENTS=, and PAGE= create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame.

```
ods html body='odspersist-body.htm'
        frame='odspersist-frame.htm'
     contents='odspersist-contents.htm'
         page='odspersist-page.htm'
;
```

The ODS SHOW statement writes to the SAS log the overall list, which, by default, is set to SELECT ALL. (See [1] in Output 3.1 on page 90.)

```
ods show;
```

The ODS SELECT statement determines which output objects ODS sends to the Listing and HTML destinations. In this case, ODS sends all output objects that are named **ParameterEstimates** and all output objects that are labelled **Type III Model ANOVA** to the two destinations.

```
ods select ParameterEstimates
           "Type III Model ANOVA";
```

The ODS SHOW statement writes to the SAS log the overall selection list, which now contains the two items that were specified in the ODS SELECT statement. (See [2] in Output 3.1 on page 90.)

```
ods show;
```

As PROC GLM sends each output object to the Output Delivery System, ODS sends the two output objects from PROC GLM that match the items in the selection list to the open destinations. (See the table of contents in Display 3.3 on page 91. Note that it is the label of an output object, not its name, that appears in the table of contents. The label for ParameterEstimates is “Solution”.)

```
proc glm data=iron;
   model loss=fe;
   title 'Parameter Estimates and Type III Model ANOVA';
run;
```

PROC GLM supports run-group processing. Therefore, the RUN statement does not end the procedure, and ODS does not automatically modify the selection list. (See [3] in Output 3.1 on page 90.)

```
ods show;
```

The QUIT statement ends the procedure. ODS removes all objects that are not specified with PERSIST from the selection list. Because this removes all objects from the list, ODS sets the list to its default, SELECT ALL.

```
quit;
```

The ODS SHOW statement writes the current selection list to the SAS log. (See [4] in Output 3.1 on page 90.)

```
ods show;
```

As PROC GLM sends each output object to the Output Delivery System, ODS sends all the output objects to the HTML and Listing destinations. (See the table of contents in Display 3.3 on page 91.)

The QUIT statement ends the procedure. Because the list uses the argument ALL, ODS does not automatically modify it when the procedure step ends.

```
proc glm data=iron;
   model loss=fe;
   title 'All Output Objects Selected';
run;
quit;
```

This ODS SELECT statement modifies the overall selection list so that it sends all output objects that are named **OverallANOVA** and all output objects that are labelled **Fit Statistics** to both the HTML and Listing destinations. The PERSIST option specifies that **OverallANOVA** should remain in the selection list when ODS automatically modifies it.

```
ods select OverallANOVA(persist) "Fit Statistics";
```

As PROC GLM sends each output object to the Output Delivery System, ODS sends the two output objects from PROC GLM that match the items in the selection list to the HTML and Listing destinations. (See the table of contents in Display 3.3 on page 91.)

```
proc glm data=iron;
   model loss=fe;
   title 'OverallANOVA and Fitness Statistics';
run;
```

When the QUIT statement ends the procedure, ODS automatically modifies the selection list. Because **OverallANOVA** was specified with the PERSIST option, it remains in the selection list. Because **Fitness Statistics** was not specified with the PERSIST option, ODS removes it from the selection list.

```
quit;
```

The ODS SHOW statement writes the current selection list to the SAS log. (See [5] in Output 3.1 on page 90.)

```
ods show;
```

As PROC GLM sends each output object to the Output Delivery System, ODS sends only the output object that is named **OverallANOVA** to the HTML and Listing destinations. (See the table of contents in Display 3.3 on page 91.)

```
proc glm data=iron;
   model loss=fe;
   title 'OverallANOVA';
   title2 'Part of the Selection List Persists';
```

```
run;
```

When the QUIT statement ends the procedure, ODS automatically modifies the selection list. Because **OverallANOVA** was specified with the PERSIST option, it remains in the selection list.

```
quit;
```

PROC PRINT does not produce any output that is named **OverallANOVA**. Therefore, no PROC PRINT output goes to the ODS destinations.

```
proc print data=iron;
   title ’The IRON Data Set’;
run;
```

This ODS SELECT statement resets all selection lists to their defaults.

```
ods select all;
```

As PROC PLOT sends each output object to the Output Delivery System, ODS sends each one to the HTML and Listing destinations because their lists and the overall list are set to SELECT ALL (the defaults).

```
proc plot data=iron;
   plot fe*loss=’*’ / vpos=25 ;
   label fe=’Iron Content’
       loss=’Weight Loss’;
   title ’Plot of Iron Versus Loss’;
run;
```

The QUIT statement ends the PLOT procedure. Because the list uses the argument ALL, ODS does not automatically modify the list when the procedure step ends.

```
quit;
```

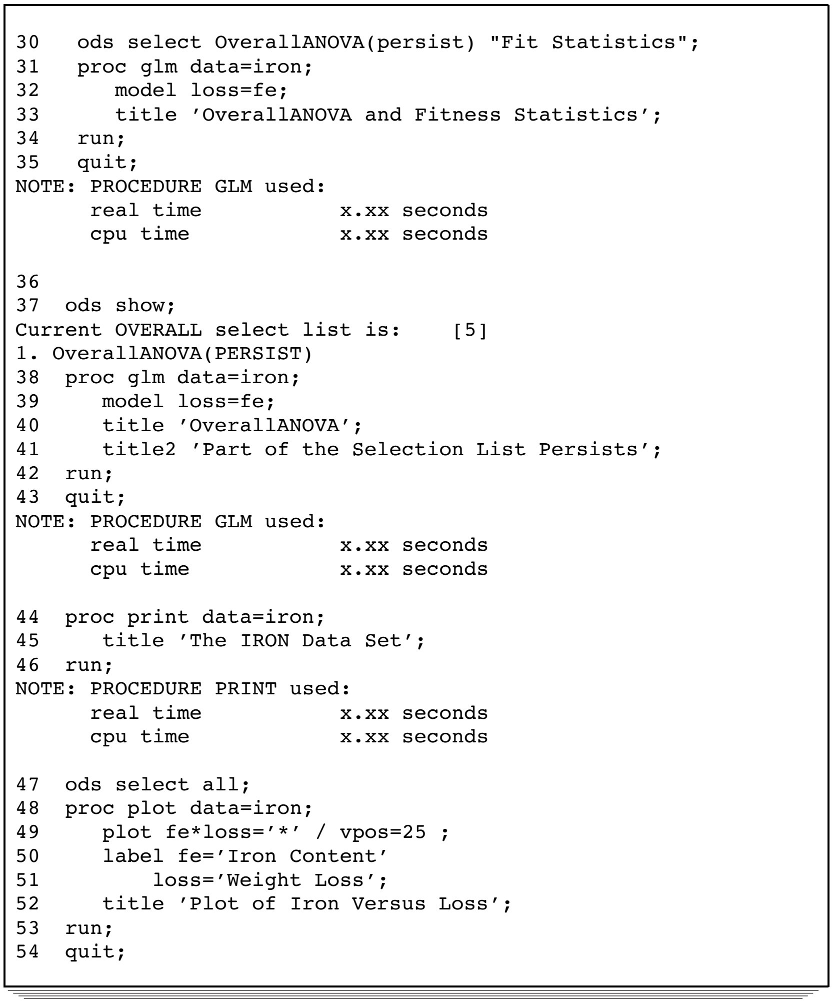
This ODS HTML statement closes the HTML destination and all the files that are associated with it.

```
ods html close;
```

SAS Log

Output 3.1 The ODS SHOW Statement Writes the Current Selection List to the SAS Log.

```
10   ods html body='odspersist-body.htm'
11        contents='odspersist-contents.htm'
12           frame='odspersist-frame.htm'
13            page='odspersist-page.htm';
NOTE: Writing HTML Body file: odspersist-body.htm
NOTE: Writing HTML Contents file: odspersist-contents.htm
NOTE: Writing HTML Pages file: odspersist-page.htm
NOTE: Writing HTML Frames file: odspersist-frame.htm
14   ods show;
Current OVERALL select list is: ALL   [1]
15   ods select ParameterEstimates
16              "Type III Model ANOVA";
17   ods show;
Current OVERALL select list is:    [2]
1. ParameterEstimates
2. "Type III Model ANOVA"
18   proc glm data=iron;
19      model loss=fe;
20      title 'Parameter Estimates and Type III Model ANOVA';
21   run;
22   ods show;
Current OVERALL select list is:    [3]
1. ParameterEstimates
2. "Type III Model ANOVA"
23   quit;
NOTE: PROCEDURE GLM used:
      real time           x.xx seconds
      cpu time            x.xx seconds

24   ods show;
Current OVERALL select list is: ALL   [4]
25   proc glm data=iron;
26      model loss=fe;
27      title 'All Output Objects Selected';
28   run;
29   quit;
NOTE: PROCEDURE GLM used:
      real time           x.xx seconds
      cpu time            x.xx seconds
```



```
30   ods select OverallANOVA(persist) "Fit Statistics";
31   proc glm data=iron;
32      model loss=fe;
33      title 'OverallANOVA and Fitness Statistics';
34   run;
35   quit;
NOTE: PROCEDURE GLM used:
      real time           x.xx seconds
      cpu time            x.xx seconds

36
37  ods show;
Current OVERALL select list is:    [5]
1. OverallANOVA(PERSIST)
38  proc glm data=iron;
39     model loss=fe;
40     title 'OverallANOVA';
41     title2 'Part of the Selection List Persists';
42  run;
43  quit;
NOTE: PROCEDURE GLM used:
      real time           x.xx seconds
      cpu time            x.xx seconds

44  proc print data=iron;
45     title 'The IRON Data Set';
46  run;
NOTE: PROCEDURE PRINT used:
      real time           x.xx seconds
      cpu time            x.xx seconds

47  ods select all;
48  proc plot data=iron;
49     plot fe*loss='*' / vpos=25 ;
50     label fe='Iron Content'
51         loss='Weight Loss';
52     title 'Plot of Iron Versus Loss';
53  run;
54  quit;
```

HTML Output

Display 3.3 Contents File Produced by the ODS HTML Statement

The contents file shows which output objects from each procedure were sent to the open ODS destinations. You can see that no output was written to the HTML destination for PROC PRINT (because it did not produce anything whose name matched the name in the selection list). You can also see that the PROC PLOT output was written to the HTML destination after the ODS SELECT ALL statement was executed.

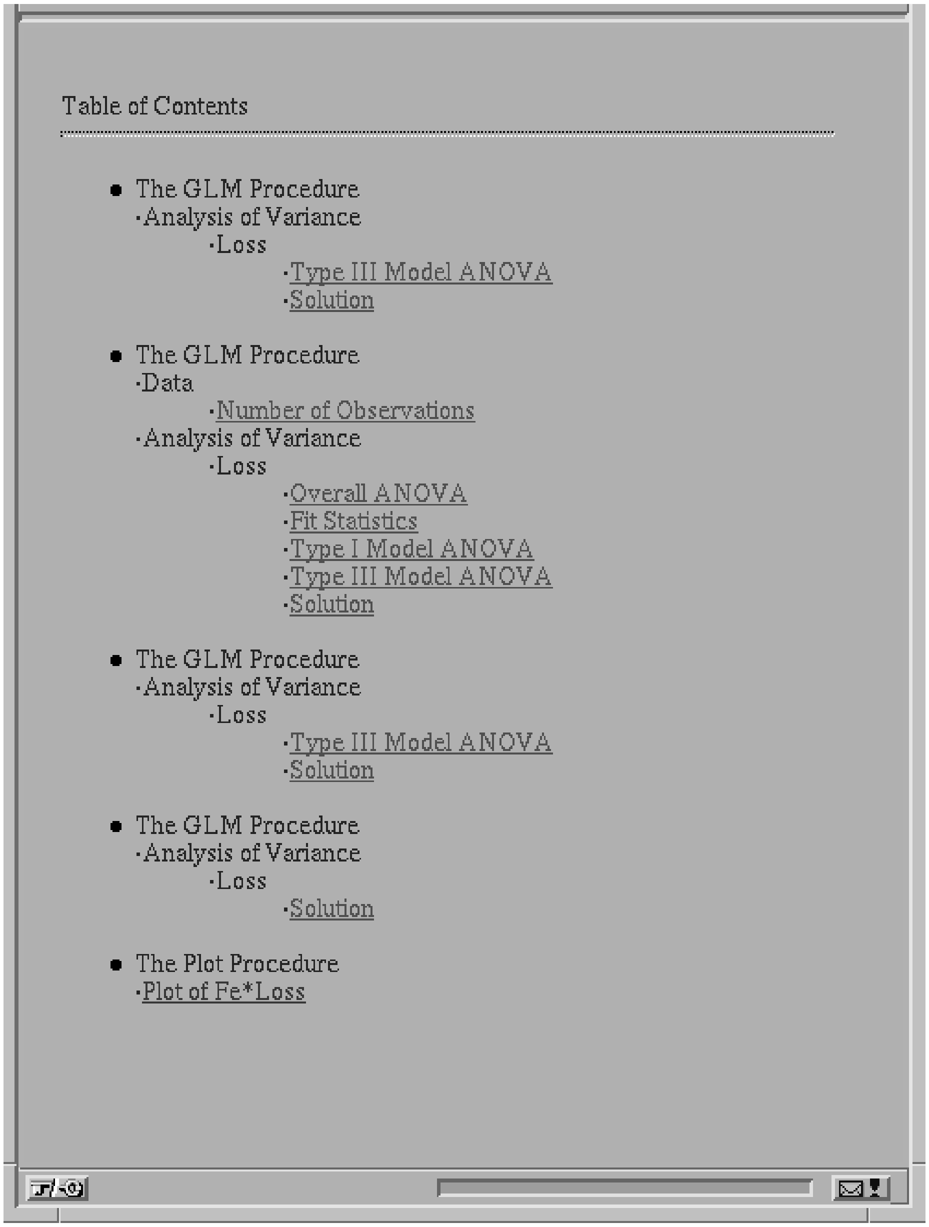


Table of Contents

- The GLM Procedure
  - ·Analysis of Variance
    - ·Loss
      - ·Type III Model ANOVA
      - ·Solution
- The GLM Procedure
  - ·Data
    - ·Number of Observations
  - ·Analysis of Variance
    - ·Loss
      - ·Overall ANOVA
      - ·Fit Statistics
      - ·Type I Model ANOVA
      - ·Type III Model ANOVA
      - ·Solution
- The GLM Procedure
  - ·Analysis of Variance
    - ·Loss
      - ·Type III Model ANOVA
      - ·Solution
- The GLM Procedure
  - ·Analysis of Variance
    - ·Loss
      - ·Solution
- The Plot Procedure
  - ·Plot of Fe*Loss

Example 5: Creating a Separate Body File for Each Page of Output

ODS features:

ODS HTML statement:

BASE=
CLOSE
NEWFILE=

Other SAS features:

PROC FORMAT

PROC SORT

PROC TABULATE

NOBYLINE|BYLINE system option

#BYVAL parameter in titles

This example creates a separate HTML file for each page of procedure output. The table of contents and table of pages do not appear any different or behave any differently from those that would be created if all the output were in a single file. Because the output is in separate files, you cannot scroll from one page of output to the next. However, you can select individual HTML files to include in a report.

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, "Alternative ODS HTML Statements for Running Examples in Different Operating Environments," on page 275. △

Program

The data set GRAIN_PRODUCTION contains data on the amounts of wheat, rice, and corn that five leading grain-producing nations produced during 1995 and 1996.

```
data grain_production;
   length Country $ 3 Type $ 5;
   input Year country $ type $ Kilotons;
   datalines;
1995 BRZ  Wheat    1516
1995 BRZ  Rice     11236
1995 BRZ  Corn     36276
1995 CHN  Wheat    102207
1995 CHN  Rice     185226
1995 CHN  Corn     112331
1995 IND  Wheat    63007
1995 IND  Rice     122372
1995 IND  Corn     9800
1995 INS  Wheat    .
1995 INS  Rice     49860
1995 INS  Corn     8223
1995 USA  Wheat    59494
1995 USA  Rice     7888
```

```
1995 USA  Corn     187300
1996 BRZ  Wheat    3302
1996 BRZ  Rice     10035
1996 BRZ  Corn     31975
1996 CHN  Wheat    109000
1996 CHN  Rice     190100
1996 CHN  Corn     119350
1996 IND  Wheat    62620
1996 IND  Rice     120012
1996 IND  Corn     8660
1996 INS  Wheat    .
1996 INS  Rice     51165
1996 INS  Corn     8925
1996 USA  Wheat    62099
1996 USA  Rice     7771
1996 USA  Corn     236064
;
```

PROC SORT sorts the data first by values of Year, then by values of Country, and finally by values of Type.

```
proc sort data=grain_production;
   by year country type;
run;
```

PROC FORMAT creates a format for the variable Country.

```
proc format;
   value $cntry 'BRZ'='Brazil'
                'CHN'='China'
                'IND'='India'
                'INS'='Indonesia'
                'USA'='United States';
run;
```

The ODS LISTING statement closes the Listing destination to conserve resources. If the destination were left open, ODS would produce both Listing and HTML output.

```
ods listing close;
```

The ODS HTML statement opens the HTML destination and creates HTML output. FRAME=, CONTENTS=, and PAGE= create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame. BASE= specifies a string to use as the first part of all links and references to the HTML files. Because no URL is specified for individual files, the final part of the link will match the filename.

```
ods html file='grain-body.htm'
     contents='grain-contents.htm'
        frame='grain-frame.htm'
         page='grain-page.htm
         base='http://www.yourcompany.com/local-address/'
```

NEWFILE=PAGE starts a new body file for each page of output. The name of each new file is based on the name specified with the FILE= option. The files that are created in this example are **grain-body.htm**, **grain-body1.htm**, **grain-body2.htm**, and **grain-body3.htm**. Because no URL is specified in FILE=, the last part of each link is taken from the filename. Therefore, as the filenames increment, so do the references to them.

```
      newfile=page;
```

The NOBYLINE option suppresses the default BY line. The #BYVAL specification inserts the value of the BY variable Year into the title.

```
options nobyline;
title 'Leading Grain-Producing Countries';
title2 'for #byval(year)';
```

This PROC REPORT step produces a report on grain production. Each BY group produces a page of output, so ODS starts a new body file for each one.

```
proc report data=grain_production nowindows
     headline headskip;
   by year;
   column country type kilotons;
   define country  / group width=14 format=$cntry.;
   define type     / group 'Type of Grain';
   define kilotons / format=comma12.;
   footnote 'Measurements are in metric tons.';
run;
```

The BYLINE option restores the default BY line. The TITLE2 statement clears the second TITLE statement.

```
options byline;
title2;
```

The TABLE statement in this PROC TABULATE step has Year as the page dimension. Therefore, PROC TABULATE explicitly produces one page of output for 1995 and one for 1996. ODS starts a new body file for each page.

```
proc tabulate data=grain_production format=comma12.;
   class year country type;
   var kilotons;
   table year,
         country*type,
         kilotons*sum=' ' / box=_page_ misstext='No data';
   format country $cntry.;
   footnote 'Measurements are in metric tons.';
run;
```

This ODS HTML statement closes the HTML destination and all the files that are associated with it. The ODS listing statement opens the Listing destination to leave the system in its default state.

```
ods html close;
ods listing;
```

HTML Output

This frame file shows the first body file. Links in the table of contents and the table of pages point to the other body files.

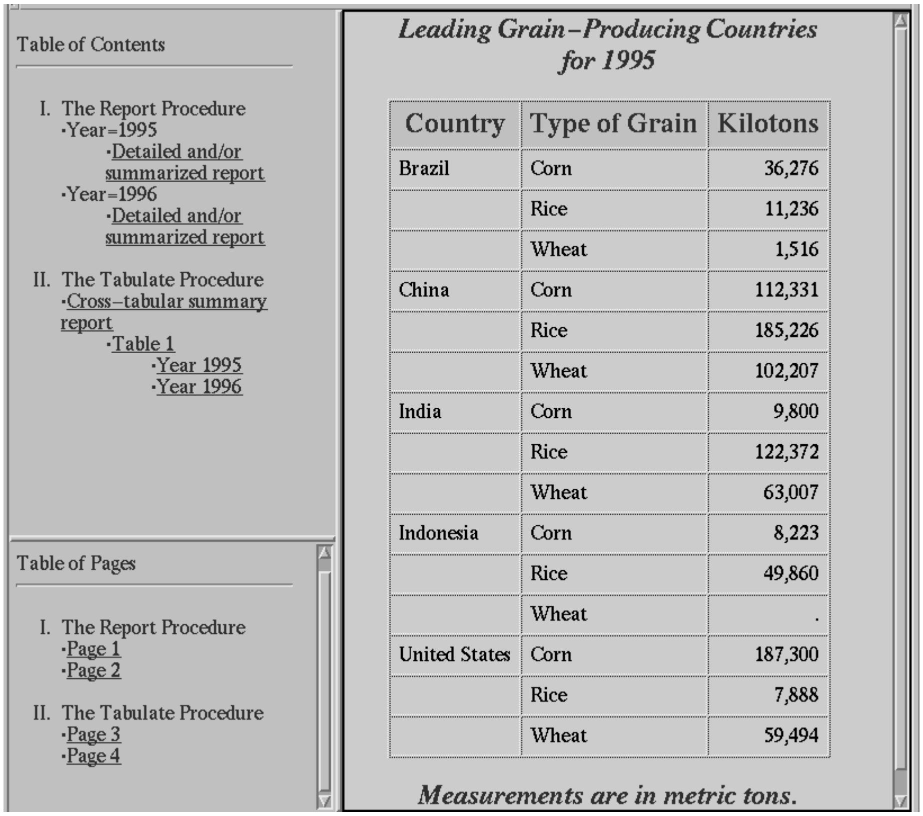


Table of Contents

I. The Report Procedure
- ·Year=1995
  - ·Detailed and/or summarized report
- ·Year=1996
  - ·Detailed and/or summarized report

II. The Tabulate Procedure
- ·Cross–tabular summary report
  - ·Table 1
    - ·Year 1995
    - ·Year 1996

Table of Pages

I. The Report Procedure
- ·Page 1
- ·Page 2

II. The Tabulate Procedure
- ·Page 3
- ·Page 4

Leading Grain–Producing Countries for 1995

Country	Type of Grain	Kilotons
Brazil	Corn	36,276
	Rice	11,236
	Wheat	1,516
China	Corn	112,331
	Rice	185,226
	Wheat	102,207
India	Corn	9,800
	Rice	122,372
	Wheat	63,007
Indonesia	Corn	8,223
	Rice	49,860
	Wheat	.
United States	Corn	187,300
	Rice	7,888
	Wheat	59,494

Measurements are in metric tons.

Links Created in the HTML Output

These HREF= attributes from the links in the contents file point to the HTML tables that ODS creates from the PROC REPORT and PROC TABULATE steps.

```
HREF="http://www.yourcompany.com/local-address/grain-body.htm#IDX"
HREF="http://www.yourcompany.com/local-address/grain-body1.htm#IDX1"
HREF="http://www.yourcompany.com/local-address/grain-body2.htm#IDX2"
HREF="http://www.yourcompany.com/local-address/grain-body3.htm#IDX3"
```

Notice how these HREF attributes are constructed:

- □ The value of the BASE= option provides the first part of the HREF, which is http://*www.yourcompany.com/local-address/*. This part of the HREF is the same for all the links that ODS creates.
- □ The value of the BODY= option, **grain-body**, provides the basis for the next part of the HREF. However, because NEWFILE= creates a new file for each output object, ODS increments this base value each time that it creates a file. The resulting file names become part of the HREF. They are **grain-body.htm**, **grain-body1.htm**, **grain-body2.htm**, and **grain-body3.htm**.
- □ The value of the ANCHOR= option provides the basis for the last part of the HREF, which follows the pound sign (#). Because the ANCHOR= option is not used in this example, ODS uses the default value of IDX. With each use, ODS increments the value of the anchor.

Example 6: Creating Data Sets from Similar Output Objects from Different Procedures

ODS features:
ODS OUTPUT statement:

MATCH_ALL
PERSIST=PROC

Other SAS features:
PROC GLM

PROC REG

PROC PRINT

Data set: IRON on page 86

This example creates and prints data sets from the parameter estimates that PROC REG and PROC GLM generate. These procedures are part of SAS/STAT software.

Note: This example uses file names that may not be valid in all operating environments. To successfully run the example in your operating environment, you may need to change the file specifications. See Appendix 1, “Alternative ODS HTML Statements for Running Examples in Different Operating Environments,” on page 275. △

Program

The OPTIONS statement controls several aspects of the Listing output. None of these options affects the HTML output.

```
options nodate pageno=1 pagesize=60 linesize=72;
```

The ODS HTML statement opens the HTML destination and creates HTML output. FRAME= and CONTENTS= create a frame that includes a table of contents that links to the contents of the body file. The body file also appears in the frame.

```
ods html body='parameter-estimates-body.htm'
        frame='parameter-estimates-frame.htm'
     contents='parameter-estimates-contents.htm';
```

The ODS SELECT statement specifies that output objects named **ParameterEstimates** should be sent to all open ODS destinations that do not specifically exclude them. The Listing destination is open by default, and its default list is SELECT ALL. The ODS HTML statement has opened the HTML destination, and its default list is also SELECT ALL. Thus any object that is named **ParameterEstimates** will go to both these destinations. The PERSIST option specifies that **ParameterEstimates** should remain in the overall selection list until the list is explicitly modified.

```
ods select ParameterEstimates(persist);
```

The ODS OUTPUT statement opens the Output destination. By default, the list for the Output destination is EXCLUDE ALL. This ODS OUTPUT statement puts **ParameterEstimates** in the selection list for the destination.

The MATCH_ALL option produces a SAS data set for each instance of **ParameterEstimates**. PERSIST=PROC ensures that the data set definition endures even when the procedure step ends. The name of the first data set is **IronParameterEstimates**. The name of the second data set is **IronParameterEstimates1**, and so on.

```
ods output ParameterEstimates(match_all persist=proc)
           =IronParameterEstimates;
```

PROC REG and PROC GLM each produce an output object named **ParameterEstimates**. Because the data set definition persists when the procedure ends, ODS creates a data set from each one.

```
proc reg data=iron;
   model loss=fe;
title 'Parameter Estimate from PROC REG';
run;
quit;

proc glm data=iron;
   model loss=fe;
title 'Parameter Estimate from PROC GLM';
run;
quit;
```

The ODS SELECT ALL statement sets the lists for all destinations to their defaults so that ODS sends all output objects to the HTML and Listing destinations. Without this statement, none of the output objects from the following PROC PRINT steps would be sent to the open destinations.

```
ods select all;
```

The PROC PRINT steps print the data sets that ODS created from PROC REG and PROC GLM. The output from these steps goes to both the HTML and the Listing destinations. Links to the output are added to the contents file.

```
proc print data=IronParameterEstimates noobs;
title 'PROC PRINT Report of the Data set from PROC REG';
run;

proc print data=IronParameterEstimates1 noobs heading=horizontal;
title 'PROC PRINT Report of the Data set from PROC GLM';
run;
```

The ODS HTML statement closes the HTML destination and all the files that are associated with it. The ODS OUTPUT statement closes the Output destination.

```
ods html close;
ods output close;
```

HTML Output

The HTML output includes the parameter estimates from PROC REG, the parameter estimates from PROC GLM, and reports of the data sets that ODS created from each set of parameter estimates.

The table of contents identifies output objects by their labels. The label for **`ParameterEstimates`** in PROC REG is **`Parameter Estimates`**. The corresponding label in PROC GLM is **`Solutions`**. Notice how the column widths in the HTML output are automatically adjusted to fit the data. Compare this layout to the layout of the columns in Output 3.2 on page 99.

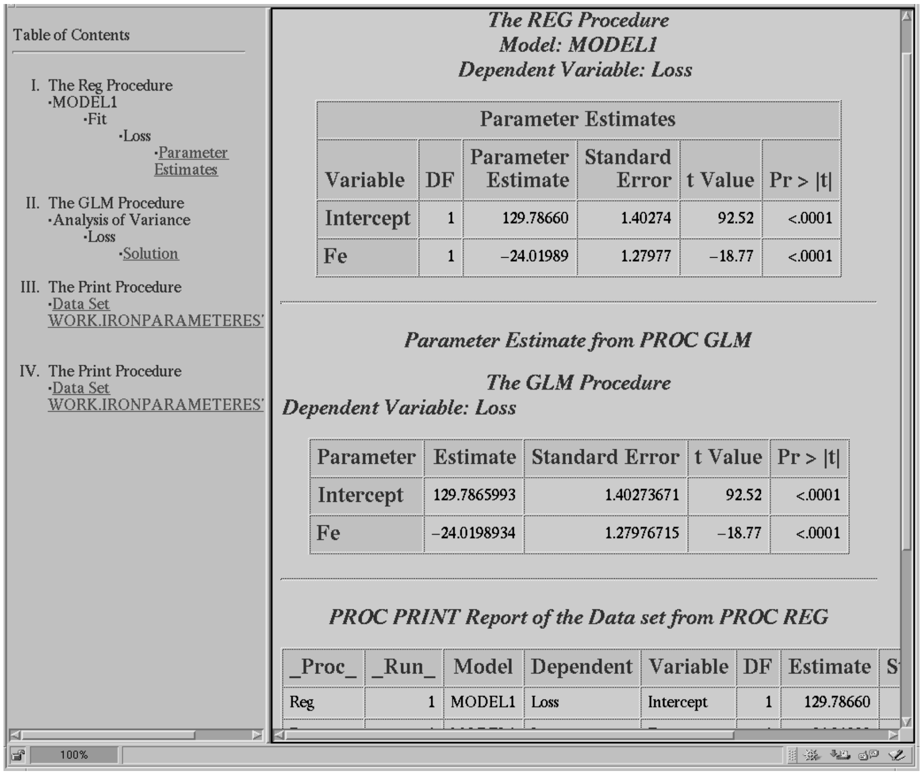


Table of Contents

I. The Reg Procedure
- MODEL1
  - Fit
    - Loss
      - Parameter Estimates

II. The GLM Procedure
- Analysis of Variance
  - Loss
    - Solution

III. The Print Procedure
- Data Set WORK.IRONPARAMETERES'

IV. The Print Procedure
- Data Set WORK.IRONPARAMETERES'

The REG Procedure
Model: MODEL1
Dependent Variable: Loss

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	129.78660	1.40274	92.52	<.0001
Fe	1	−24.01989	1.27977	−18.77	<.0001

Parameter Estimate from PROC GLM

The GLM Procedure
Dependent Variable: Loss

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	129.7865993	1.40273671	92.52	<.0001
Fe	−24.0198934	1.27976715	−18.77	<.0001

PROC PRINT Report of the Data set from PROC REG

Proc	_Run_	Model	Dependent	Variable	DF	Estimate	S
Reg	1	MODEL1	Loss	Intercept	1	129.78660	

Listing Output

Output 3.2 Listing Output from the REG, GLM, and PRINT Procedures

```
                    Parameter Estimate from PROC REG                   1

                           The REG Procedure
                             Model: MODEL1
                        Dependent Variable: Loss

                           Parameter Estimates

                        Parameter       Standard
Variable      DF         Estimate          Error     t Value    Pr > |t|

Intercept      1        129.78660        1.40274       92.52      <.0001
Fe             1        -24.01989        1.27977      -18.77      <.0001
```

```
                    Parameter Estimate from PROC GLM                   2

                           The GLM Procedure

Dependent Variable: Loss

                                     Standard
   Parameter         Estimate           Error    t Value    Pr > |t|

   Intercept      129.7865993      1.40273671      92.52      <.0001
   Fe             -24.0198934      1.27976715     -18.77      <.0001
```

```
            PROC PRINT Report of the Data set from PROC REG            3

       _Proc_    _Run_    Model     Dependent    Variable     DF

        Reg          1    MODEL1      Loss       Intercept     1
        Reg          1    MODEL1      Loss       Fe            1

          Estimate         StdErr     tValue     Probt

         129.78660        1.40274      92.52    <.0001
         -24.01989        1.27977     -18.77    <.0001
```

```
            PROC PRINT Report of the Data set from PROC GLM            4

       _Proc_    _Run_    Dependent    Parameter        Estimate

        GLM          1      Loss       Intercept     129.7865993
        GLM          1      Loss       Fe            -24.0198934

             StdErr     tValue     Probt

         1.40273671      92.52    <.0001
         1.27976715     -18.77    <.0001
```

Example 7: Appending to HTML Files

ODS features:

ODS HTML statement:

ANCHOR=

BODY= option with a fileref

NO_BOTTOM_MATTER suboption

STYLE=

Other SAS features:

FILENAME statement

PROC PRINT

PROC REPORT

DATA _null_ statement

Data set: GRAIN_PRODUCTION on page 93

Format: $CNTRY. on page 94

This example creates HTML output from PROC PRINT and PROC REPORT. It also uses the DATA step to write customized HTML to the file that contains the HTML output. The DATA step executes between procedure steps.

Program

The FILENAME statement assigns a fileref to the file that will contain the HTML output.

```
filename reports 'external-file';
```

The ODS LISTING statement closes the Listing destination to conserve resources. If the destination were left open, ODS would produce both Listing and HTML output.

```
ods listing close;
```

The ODS HTML statement opens the HTML destination and creates HTML output. NO_BOTTOM_MATTER suppresses the writing of the HTML that, by default, ODS writes at the end of a file.

```
ods html body=reports (no_bottom_matter)
```

STYLE= specifies the style definition to use in writing the HTML output.

```
         style=D3D;
```

This PROC PRINT step prints the observations in the data set GRAIN_PRODUCTION that have a value of 1996 for the variable Year.

```
proc print data=grain_production;
   var country type kilotons;
   format country $cntry. kilotons comma12.;
   where year=1996;
   title 'Leading Grain-Producing Countries';
   footnote 'Measurements are in metric tons.';
run;
```

The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it.

```
ods html close;
```

This FILENAME statement assigns a fileref to the file that will contain the HTML output. MOD is a host-specific option for the UNIX operating environment that opens the file in update mode.

```
filename reports 'external-file' mod;
```

This DATA step writes to the file that is referenced by REPORTS. The PUT statements create an H1 header in the HTML file.

```
data _null_;
   file reports;
   put '<h1>The preceding output is from PROC PRINT.';
   put 'I am going to try a variety of procedures.';
   put 'Let me know which procedure you prefer.';
   put 'By the way, this report uses the D3D style.</h1>';
run;
```

This ODS HTML statement opens the HTML destination and creates HTML output. The NO_TOP_MATTER and the NO_BOTTOM_MATTER suboptions suppress the writing of the HTML that ODS usually writes to the top and the bottom of a file.

```
ods html body=reports (no_top_matter no_bottom_matter)
```

ANCHOR= specifies **report** as the base name for the HTML anchor tags. When you use ODS to append to an HTML file that ODS created, you must specify a new anchor name each time that you open the file from ODS so that you don't write the same anchors to the file again. (ODS cannot know about anchors that are already in the file when it opens it, and by default it uses **IDX** as the base for anchor names.)

```
         anchor='report';
```

This PROC REPORT step prints the data set. ODS creates HTML output in the body file.

```
proc report data=grain_production nowindows
            headline headskip;
   where year=1996;
   column country type kilotons;
   define country / group width=14 format=$cntry.;
   define type / group 'Type of Grain';
   define kilotons / format=comma12.;
run;
```

The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it.

```
ods html close;
```

This DATA step writes to the file that is referenced by REPORTS. The PUT statements create an H1 header in the HTML file.

```
data _null_;
   file reports;
   put ''<h1>The preceding output is from PROC REPORT.'';
   put ''It doesn't repeat the name of the country on every line.'';
   put ''This report uses the default style.</h1>'';
run;
```

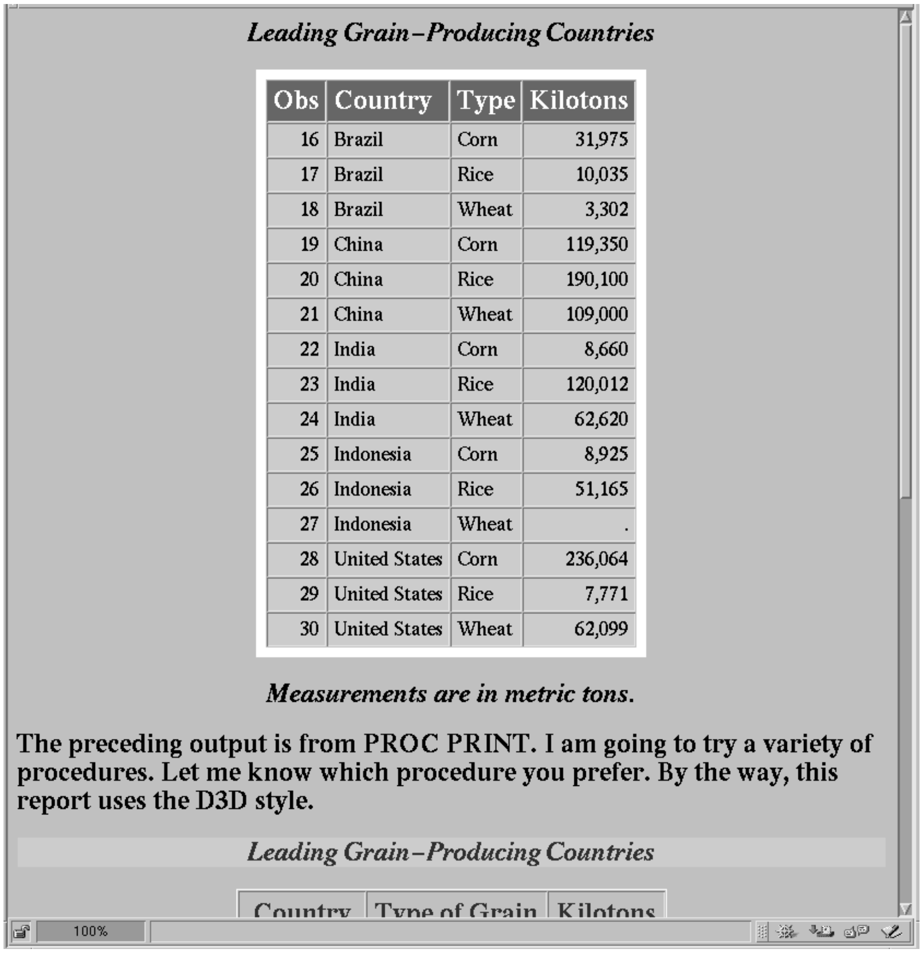
In order to write the bottom matter to the HTML file so that it contains valid HTML, you must open the HTML destination one more time. NO_TOP_MATTER ensures that the top matter is not placed in the file again. ANCHOR= provides a new base name for the anchors in the bottom matter.

```
ods html file=reports(no_top_matter) anchor='end';
```

The ODS HTML CLOSE statement closes the HTML destination and all the files that are open for that destination.

```
ods html close;
```

HTML Output



Leading Grain-Producing Countries

Obs	Country	Type	Kilotons
16	Brazil	Corn	31,975
17	Brazil	Rice	10,035
18	Brazil	Wheat	3,302
19	China	Corn	119,350
20	China	Rice	190,100
21	China	Wheat	109,000
22	India	Corn	8,660
23	India	Rice	120,012
24	India	Wheat	62,620
25	Indonesia	Corn	8,925
26	Indonesia	Rice	51,165
27	Indonesia	Wheat	.
28	United States	Corn	236,064
29	United States	Rice	7,771
30	United States	Wheat	62,099

Measurements are in metric tons.

The preceding output is from PROC PRINT. I am going to try a variety of procedures. Let me know which procedure you prefer. By the way, this report uses the D3D style.

Leading Grain-Producing Countries

Country	Type of Grain	Kilotons

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *The Complete Guide to the SAS® Output Delivery System, Version 8,* Cary, NC: SAS Institute Inc., 1999. 310 pp.

The Complete Guide to the SAS® Output Delivery System, Version 8

ISBN 1–58025–425–X

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.