CHAPTER

16

SAS Output

Definitions

SAS output is the result of executing SAS programs. Most SAS procedures and some DATA step applications produce output.

There are two types of SAS output:

SAS log
contains a description of the SAS session and lists the lines of source code that were executed. Depending on the setting of SAS system options, the method of running SAS, and the program statements that you specify, the log can include the following types of information:

- program statements
- names of data sets created by the program
- notes, warnings, or error messages encountered during program execution
- the number of variables and observations each data set contains
- processing time required for each step.

You can write specific information to the SAS log (such as variable values or text strings) by using the SAS statements that are described in “Writing to the Log” on page 201.

The log is also used by some of the SAS procedures that perform utility functions, for example the DATASETS and OPTIONS procedures. See the *SAS Procedures Guide* for more information.

Because the SAS log provides a journal of program processing, it is an essential debugging tool. However, certain system options must be in effect to make the log effective for debugging your SAS programs. “Customizing the Log” on page 201 describes several SAS system options that you can use.

program results

contain the results of most SAS procedures and some DATA step applications. Results can be routed to a file, and printed as a listing. If you use the Output Delivery System (ODS), you can produce results for a high resolution printer or create HTML output for use with a web browser. You can customize your output by modifying or creating your own table definitons, which are descriptions of how you want to format your output. For more information about the flexibility of ODS, see *The Complete Guide to the SAS Output Delivery System*.

Routing SAS Output

The destination of your output depends on

- the operating environment
- the setting of SAS system options
- whether you use ODS
- the method of running SAS.

There are several ways to route SAS output to a destination other than the default destination. You can route the output to your terminal, to an external file, or directly to a printer. If you use ODS, you can route output to a data set for most procedures.

Operating Environment Information: See the SAS documentation for your operating environment for specific information. △

The following table shows the default destination of SAS output for each method of operation.

Table 16.1 Default Destinations of SAS Output

Method of Running SAS	Destination of SAS Output
windowing mode (Explorer window)	the Log window or the output window
interactive line mode	the terminal display (as statements are entered)
noninteractive mode	depends on the operating environment
batch mode	depends on the operating environment

Operating Environment Information: The default destination for SAS output is specific to your operating environment. For specific information, see the SAS documentation for your operating environment. △

The SAS Log

Structure of the Log

The SAS log is a record of everything you do in your SAS session or with your SAS program. Original program statements are identified by line numbers. Interspersed with SAS statements are messages from SAS. These messages may begin with the words NOTE, INFO, WARNING, ERROR, or an error number, and they may refer to a SAS statement by its line number in the log.

For example, in the following output, the number 1 prints to the left of the OPTIONS statement. This means that it is the first line in the program. In interactive mode, SAS continues with the sequence of line numbering until you end your session. If you submit the program again (or submit other programs in your current SAS session), the first program line number will be the next consecutive number.

Operating Environment Information: The SAS log appears differently depending on your operating environment. See the SAS documentation for your operating environment. △

Output 16.1 Sample SAS Log

```
NOTE: Copyright (c) 1999 by SAS Institute Inc., Cary, NC, USA.  ❶
NOTE: SAS (r) Proprietary Software Version 8 (TS00.00P1D06081999) ❷
      Licensed to SAS Institute Inc., Site 0000000001.  ❸
NOTE: This session is executing on the HP-UX B.10.20 platform.  ❹

NOTE: Running on HP Model 9000/782 Serial Number 2011922219.  ❺


NOTE: SAS initialization used:
      real time           4.20 seconds
      cpu time            1.18 seconds

1    options pagesize=24 linesize=64 nodate;  ❻
2
3    data logsample;  ❼
4       infile '/u/abcdef/testdir/sampledata.dat';
5       input LastName $ 1-12 ID $ Gender $ Birth : date7.  ❽
5  ! score1 score2 score3
6                                                          score
6  ! 4 score5 score6;
7       format Birth mmddyy8.;
8    run;

NOTE: The infile '/u/abcdef/testdir/sampledata.dat' is:  ❾
      File Name=/u/abcdef/testdir/sampledata.dat,
      Owner Name=abcdef,Group Name=pubs,
      Access Permission=rw-r--r--,
      File Size (bytes)=296

NOTE: 5 records were read from the infile  ❿
      '/u/abcdef/testdir/sampledata.dat'.
      The minimum record length was 58.
      The maximum record length was 59.
NOTE: The data set WORK.LOGSAMPLE has 5 observations and 10
      variables.  ⓫
NOTE: DATA statement used:
      real time           0.44 seconds
      cpu time            0.13 seconds


9
10   proc sort data=logsample;  ⓬
11      by LastName;
12

NOTE: There were 5 observations read from the dataset
      WORK.LOGSAMPLE.
NOTE: The data set WORK.LOGSAMPLE has 5 observations and 10
      variables.  ⓭
NOTE: PROCEDURE SORT used:
      real time           0.16 seconds
      cpu time            0.03 seconds


13   proc print data=logsample;  ⓮
14      by LastName;
15   run;

NOTE: There were 5 observations read from the dataset
      WORK.LOGSAMPLE.
NOTE: PROCEDURE PRINT used:
      real time           0.31 seconds
      cpu time            0.05 seconds
```

The following list corresponds to the circled numbers in the SAS log shown above:

❶ copyright information.

❷ SAS system release used to run this program.

❸ name and site number of the computer installation where the program ran.

❹ platform used to run the program.

❺ hardware used to run the program.

❻ OPTIONS statement. This statement uses SAS system options to set a page size of 24 and a line size of 64, and to suppress the date in the output.

❼ SAS statements that make up the program (if the SAS system option SOURCE is enabled).

❽ long statement continued to the next line. Note that the continuation line is preceded by an exclamation point (!), and that the line number does not change.

❾ input file information-notes or warning messages about the raw data and where they were obtained (if the SAS system option NOTES is enabled).

❿ the number and record length of records read from the input file (if the SAS system option NOTES is enabled).

⓫ SAS data set that your program created; notes that contain the number of observations and variables for each data set created (if the SAS system option NOTES is enabled).

⓬ procedure that sorts your data set

⓭ note about the sorted SAS data set

⓮ procedure that prints your data set.

Writing to the Log

You can instruct SAS to write additional information to the log by using the following statements:

PUT statement
: writes selected lines (including text strings and DATA step variable values) to the SAS log. This behavior of the PUT statement requires that your program does not execute a FILE statement before the PUT statement in the current iteration of a DATA step, and that it does not execute a **`FILE LOG;`** statement.

%PUT statement
: enables you to write a text string or macro variable values to the SAS log. %PUT is a SAS macro program statement that is independent of the DATA step and can be used anywhere.

LIST statement
: writes to the SAS log the input data records for the data line that is being processed. The LIST statement operates only on data that are read with an INPUT statement. It has no effect on data that are read with a SET, MERGE, MODIFY, or UPDATE statement. Use the LIST statement in a DATA step.

ERROR statement
: sets the automatic _ERROR_ variable to 1 and optionally writes to the log a message that you specify. Use the ERROR statement in a DATA step.

Use the PUT, LIST, and ERROR statements in combination with conditional processing to debug DATA steps by writing selected information to the log.

Customizing the Log

Altering the Contents of the Log

When you have large SAS production programs or an application that you run on a regular basis without changes, you might want to suppress part of the log. SAS system options enable you to suppress SAS statements and system messages, as well as to limit the number of error messages. Note that all SAS system options remain in effect for the duration of your session or until you change the options. You should not suppress log messages until you have successfully executed the program without errors.

The following list describes some of the SAS system options that you can use to alter the contents of the log:

CPUID | NOCPUID
: controls whether hardware information is written to the SAS log.

ECHOAUTO | NOECHOAUTO
: controls whether autoexec code in an input file is written to the log.

ERRORS=*n*
: specifies the maximum number of observations for which data error messages are printed.

MPRINT | NOMPRINT
: controls whether SAS statements that are generated by macro execution are written to the SAS log.

MSGLEVEL=N | I
: controls the level of detail in messages that are written to the SAS log. If the MSGLEVEL system option is set to N, the log displays notes, warnings, and error messages only. If MSGLEVEL is set to I, the log displays additional notes pertaining to index usage, merge processing, and sort utilities, along with standard notes, warnings, and error messages.

NEWS=*external-file*
: controls whether news information that is maintained at your site is written to the SAS log.

NOTES | NONOTES
: controls whether notes (messages beginning with NOTE) are written to the SAS log. NONOTES does not suppress error or warning messages.

OVP | NOOVP
: controls whether output lines that are printed by SAS are overprinted.

PRINTMSGLIST | NOPRINTMSGLIST
: controls whether extended lists of messages are written to the SAS log.

SOURCE | NOSOURCE
: controls whether SAS writes source statements to the SAS log.

SOURCE2 | NOSOURCE2
: controls whether SAS writes secondary source statements from files included by %INCLUDE statements to the SAS log.

SYMBOLGEN | NOSYMBOLGEN
: controls whether the results of resolving macro variable references are written to the SAS log.

For more information about how to use these and other SAS system options, see “SAS System Options” in *SAS Language Reference: Dictionary*.

Operating Environment Information: See the documentation for your operating environment for other options that affect log output. △

Customizing the Appearance of the Log

The following SAS statements and SAS system options enable you to customize the log. Customizing the log is helpful when you use the log for report writing or for creating a permanent record.

DATE system option	controls whether the date and time that the SAS job began are printed at the top of each page of the SAS log and any print file created by SAS.
FILE statement	enables you to write the results of PUT statements to an external file. You can use the following two options in the FILE statement to customize the log for that report. LINESIZE=*value* specifies the maximum number of columns per line for reports and the maximum record length for data files. PAGESIZE=*value* specifies the maximum number of lines to print on each page of output. *Note:* FILE statement options apply only to the output specified in the FILE statement, whereas the LINESIZE= and PAGESIZE= SAS system options apply to all subsequent listings. △
LINESIZE= system option	specifies the line size (printer line width) for the SAS log and the SAS procedure output file that are used by the DATA step and procedures.
MISSING= system option	specifies the character to be printed for missing numeric variable values.
NUMBER system option	controls whether the page number prints on the first title line of each page of printed output.
PAGE statement	skips to a new page in the SAS log and continues printing from there.
PAGESIZE= system option	specifies the number of lines that you can print per page of SAS output.
SKIP statement	skips a specified number of lines in the SAS log.

Operating Environment Information: The range of values for the FILE statement and for SAS system options depends on your operating environment. See the SAS documentation for your operating environment for more information. △

For more information about how to use these and other SAS system options and statements, see “SAS System Options” and “Statements” in *SAS Language Reference: Dictionary*.

Traditional SAS Listing Output

Example of Traditional Listing Output

Many SAS procedures process or analyze data and can produce output as one result. You can also generate a listing by the DATA step, using a combination of the FILE and PUT statements.

See the procedure descriptions in the *SAS Procedures Guide* for examples of output from SAS procedures. For a discussion and examples of DATA step output, see the FILE and PUT statements in *SAS Language Reference: Dictionary*.

This example produces a listing that is generated by the PUT and FILE statements in a DATA step. The input file is the SAS data set GRAIN_PRODUCERS.

```
options pagesize=60 linesize=64 nodate pageno=1;

title 'Leading Grain Producers';
title2 'for 1996';

data _null_;
   set grain_producers;
   file print header=newpage;
   if year=1996;
   format country $cntry.;
   label type='Grain';
   put country @25 type @50 kilotons;
   return;
   newpage:
      put 'Country' @25 'Grain' @50 'Kilotons';
      put 60*'=';
      return;
 run;
```

```
                    Leading Grain Producers                   1
                            for 1996
Country                 Grain                    Kilotons
============================================================
Brazil                  Wheat                    3302
Brazil                  Rice                     10035
Brazil                  Corn                     31975
China                   Wheat                    109000
China                   Rice                     190100
China                   Corn                     119350
India                   Wheat                    62620
India                   Rice                     120012
India                   Corn                     8660
Indonesia               Wheat                    .
Indonesia               Rice                     51165
Indonesia               Corn                     8925
United States           Wheat                    62099
United States           Rice                     7771
United States           Corn                     236064
```

Making Output Descriptive

There are several ways to customize SAS procedure output and DATA step output. You can change the look of output by adding informative titles, footnotes, and labels, and by changing the way the information is formatted on the page. The following list describes some of the statements and SAS system options that you can use.

CENTER | NOCENTER system option
: controls whether output is centered. By default, SAS centers titles and procedure output on the page and on the terminal display.

DATE | NODATE system option
: controls printing of date and time values. When this option is enabled, SAS prints on the top of each page of output the date and time the SAS job started. When you run SAS in interactive mode, the date and time the job started is the date and time you started your SAS session.

FOOTNOTE statements
: print footnotes at the bottom of each output page. You can also use the FOOTNOTES window for this purpose.

FORMCHAR=
: specifies the default output formatting characters for some procedures such as CALENDAR, FREQ, REPORT, and TABULATE.

FORMDLIM=
: specifies a character that is used to delimit page breaks in SAS output.

LABEL statement
: associates descriptive labels with variables. With most procedure output, SAS writes the label rather than the variable name.

  The LABEL statement also provides descriptive labels when it is used with certain SAS procedures. See the *SAS Procedures Guide* for information on using the LABEL statement with a specific procedure (for example, the PRINT procedure).

LINESIZE= and PAGESIZE= system options
: can change the default number of lines per page (page size) and characters per line (line size) for printed output. The default depends on the method of running SAS and the settings of certain SAS system options. Specify new page and line sizes in the OPTIONS statement or OPTIONS window. You can also specify line and page size for DATA step output in the FILE statement.

  The values you use for the LINESIZE= and PAGESIZE= system options can significantly affect the appearance of the output that is produced by some SAS procedures.

NUMBER | NONUMBER and PAGENO= system options
: control page numbering. The NUMBER system option controls whether the page number prints on the first title line of each page of printed output. You can also specify a beginning page number for the next page of output produced by SAS by using the PAGENO= system option.

TITLE statements
: print titles at the top of each output page. By default, SAS prints the following title:

```
The SAS System
```

  You can use the TITLE statement or TITLES window to replace the default title or specify other descriptive titles for SAS programs. You can use the null title statement (**`title;`**) to suppress a TITLE statement.

For more information about how to use these and other SAS system options and statements, see “SAS System Options” and “Statements” in *SAS Language Reference: Dictionary*.

Reformatting Values

Certain SAS statements, procedures, and options enable you to print values using specified formats. You can apply or change formats with the FORMAT and ATTRIB statements, or with the VAR window in a windowing environment.

The FORMAT procedure enables you to design your own formats and informats, giving you added flexibility in displaying values. See "The FORMAT Procedure," in the *SAS Procedures Guide* for more information.

Printing Missing Values

SAS represents ordinary missing numeric values in a SAS listing as a single period, and missing character values as a blank space. If you specified special missing values for numeric variables, SAS writes the letter or the underscore. For character variables, SAS writes a series of blanks equal to the length of the variable.

The MISSING= system option enables you to specify a character to print in place of the period for ordinary missing numeric values.

Changing the Destination of the Log and the Output

You can redirect both the SAS log and procedure output to your terminal display, to a printer, or to an external file. You can redirect output using the following methods:

PRINTTO procedure
: routes DATA step, log, or procedure output from the system default destinations to the destination you choose.

FILENAME statement
: associates a fileref with an external file or output device and enables you to specify file and device attributes.

FILE command
: stores the contents of the LOG or OUTPUT windows in files you specify, when the command is issued from within the windowing environment.

SAS system options
: redefine the destination of log and output for an entire SAS program. These system options are specified when you invoke SAS. The system options used to route output are the ALTLOG=, ALTPRINT=, LOG=, and PRINT= options.

Operating Environment Information: The way you specify output destinations when you use SAS system options is dependent on your operating environment. See the SAS documentation for your operating environment for details. △

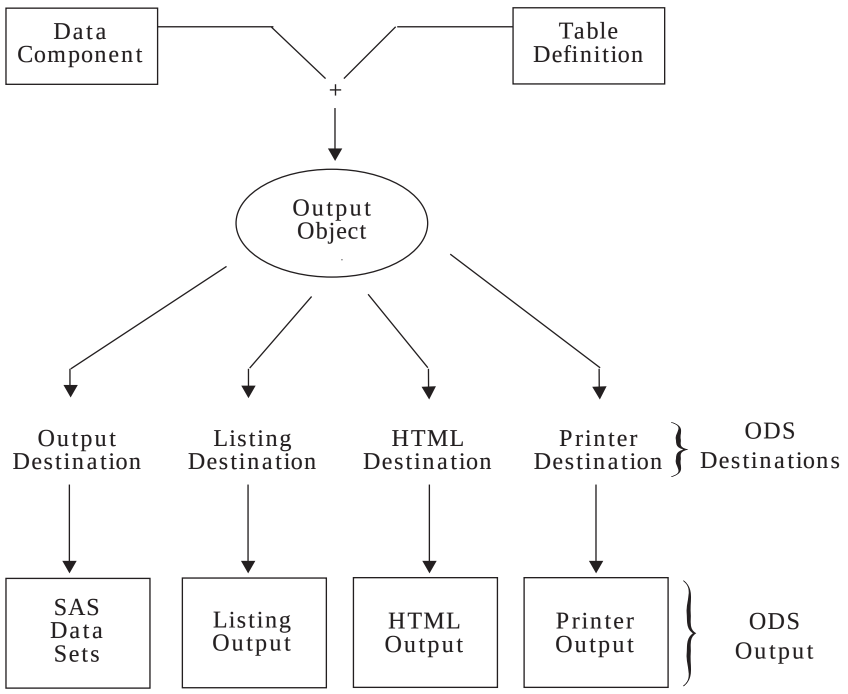
Creating Output Using the Output Delivery System (ODS)

Definition

The *Output Delivery System* (ODS) provides greater flexibility in choosing the kind of output you want to produce. Within the DATA step itself, the ODS option in the FILE statement and the _ODS_ option in the PUT statement provide connections with the Output Delivery System. You can use both of these connections to route the results of a DATA step to ODS.

In the past, the program output has generally referred to the outcome of a SAS procedure step. You could send this output to the Output window if you were working in a windowing environment, to an output device if you were working in line mode, or to a file if you used PROC PRINTTO or SAS system options. With the advent of the Output Delivery System, “output” takes on a much broader meaning. The following figure illustrates the concept of output for Version 8 SAS. Definitions of the terms in the figure follow.

Figure 16.1 Model of the Production of ODS Output



data component
: Each procedure that supports ODS and each DATA step produces one or more data components. Each data component contains results (numbers and characters) of the step in a form similar to a SAS data set.

table definition
: A table definition is a description of how to render a tabular data component. (Almost all ODS output is tabular.) This description includes but is not limited to

- □ the order of the columns
- □ text and order of column headers
- □ formats for data
- □ font sizes and font faces.

Note: Not all procedures use a table definition. △

Procedures like PROC CHART and PROC TIMEPLOT need to use a monospace font to correctly align their results. Such procedures do not use a table definition. The HTML and Printer output that they produce does not substantially differ from the Listing output. You cannot alter their HTML or Printer output except by specifying a different style definition.

Procedures like PROC PRINT, PROC REPORT, and PROC TABULATE produce an endless variety of results, depending on how you use the procedures. These procedures do not use a table definition either. However, both PROC REPORT and PROC TABULATE provide ways for you to customize their HTML and Printer output. For more information, see “Fundamental Concepts for Using Base SAS Procedures” in the *SAS Procedures Guide*, as well as the sections on PROC REPORT and PROC TABULATE procedures in the same document.

output object
: ODS binds a table definition to a data component in order to produce an output object. The output object, therefore, contains both the results of the procedure or DATA step and information about how to render the results. An output object has a name and a label.

  Note: Although many output objects include a table definition, not all do. In some cases the output object is no more than the data component. △

ODS destinations
: ODS currently supports four destinations:

  - The Listing destination produces monospace output, which is formatted like traditional SAS procedure output.
  - The HTML destination produces output that is formatted in Hypertext Markup Language.
  - The Printer destination produces output that is formatted for high-resolution printers.
  - The Output destination produces SAS data sets.

ODS output
: ODS output consists of formatted output objects from any of the ODS destinations. The Output destination produces SAS data sets. The Listing, HTML, and Printer destinations produce Listing output, HTML output, and Printer output.

Traditional SAS output is, then, one kind of ODS output (Listing output), but it is no longer the only kind. For complete information about ODS, see *The Complete Guide to the SAS Output Delivery System*.

ODS Destinations

ODS currently supports four destinations: the HTML destination, the Listing destination, the Output destination, and the Printer destination. ODS destinations can be open or closed. When a destination is open, ODS can send output objects to it. When a destination is closed, ODS cannot send output objects to it. An open destination uses system resources even if you use the selection and exclusion features of ODS to exclude all output objects from the destination. (See “Using Selection and Exclusion Lists” on page 208.) Therefore, to conserve resources, close all unnecessary destinations.

You open and close a destination with the appropriate ODS statement (ODS HTML, ODS LISTING, ODS PRINTER, or ODS OUTPUT). By default, the Listing destination is open, and all other destinations are closed. Consequently, if you do nothing, your SAS programs run and produce Listing output, just as they did in previous releases of SAS before ODS was available.

Using Selection and Exclusion Lists

For each ODS destination, ODS maintains either a selection list (a list of output objects to send to the destination) or an exclusion list (a list of output objects to exclude from the destination). ODS also maintains an overall selection list or an overall exclusion list. You can use these lists to control which output objects go to which ODS destinations.

To see the contents of the lists use the ODS SHOW statement, which writes the lists to the SAS log. The following table shows the default lists:

Table 16.2 Default List for Each ODS Destination

ODS Destination	Default List
HTML	SELECT ALL
Listing	SELECT ALL
Output	EXCLUDE ALL
Printer	SELECT ALL
Overall	SELECT ALL

Example 1: Using the Default Table Definition

This example uses the DATA step's default table definition to write an output object to the Listing destination.

```
     /* The OPTIONS statement controls several aspects of the */
     /* Listing output.                                       */
  options pagesize=60 linesize=64 nodate pageno=1;

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

     /* The DATA step does not create a data set. Instead,   */
     /* it creates a data component and, eventually, an      */
     /* output object.                                       */
  data _null_;
     length Country $ 3 Type $ 5;
     input Year country $ type $ Kilotons;
     format country $cntry.;
     label type='Grain';

        /* The combination of the fileref PRINT and the ODS   */
        /* option in the FILE statement routes the DATA step  */
        /* output to ODS. The only open ODS destination is    */
        /* the Listing destination, which is open by default. */
        /* Because no suboptions are specified, ODS uses the  */
        /* default DATA step table definition.                */
     file print ods;
```

```
         /* The _ODS_ option in the PUT statement writes all    */
         /* the variables to its buffer. Then the PUT statement */
         /* writes to the data component.                       */
      put _ods_;
      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
   ;
```

In the following output, the default table definition produces a column for each variable in the DATA step. The order of the columns is determined by their order in the program data vector. Because no attributes are specified for individual columns, ODS uses the default column headers and formats.

Output 16.2 Listing Output Created with the Default DATA Step Table Definition

```
                 Leading Grain Producers                  1
       Country       Grain        Year         Kilotons

    Brazil           Wheat            1995            1516
    Brazil           Rice             1995           11236
    Brazil           Corn             1995           36276
    China            Wheat            1995          102207
    China            Rice             1995          185226
    China            Corn             1995          112331
    India            Wheat            1995           63007
    India            Rice             1995          122372
    India            Corn             1995            9800
    Indonesia        Wheat            1995               .
    Indonesia        Rice             1995           49860
    Indonesia        Corn             1995            8223
    United States    Wheat            1995           59494
    United States    Rice             1995            7888
    United States    Corn             1995          187300
    Brazil           Wheat            1996            3302
    Brazil           Rice             1996           10035
    Brazil           Corn             1996           31975
    China            Wheat            1996          109000
    China            Rice             1996          190100
    China            Corn             1996          119350
    India            Wheat            1996           62620
    India            Rice             1996          120012
    India            Corn             1996            8660
    Indonesia        Wheat            1996               .
    Indonesia        Rice             1996           51165
    Indonesia        Corn             1996            8925
    United States    Wheat            1996           62099
    United States    Rice             1996            7771
    United States    Corn             1996          236064
```

Example 2: Selecting Variables for the Data Component

This example selects variables to write to the data component. The output is routed to two ODS destinations: the Listing destination, which is open by default, and the HTML destination, which is opened by the ODS HTML statement.

Operating Environment Information: 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 the SAS documentation for your operating environment. △

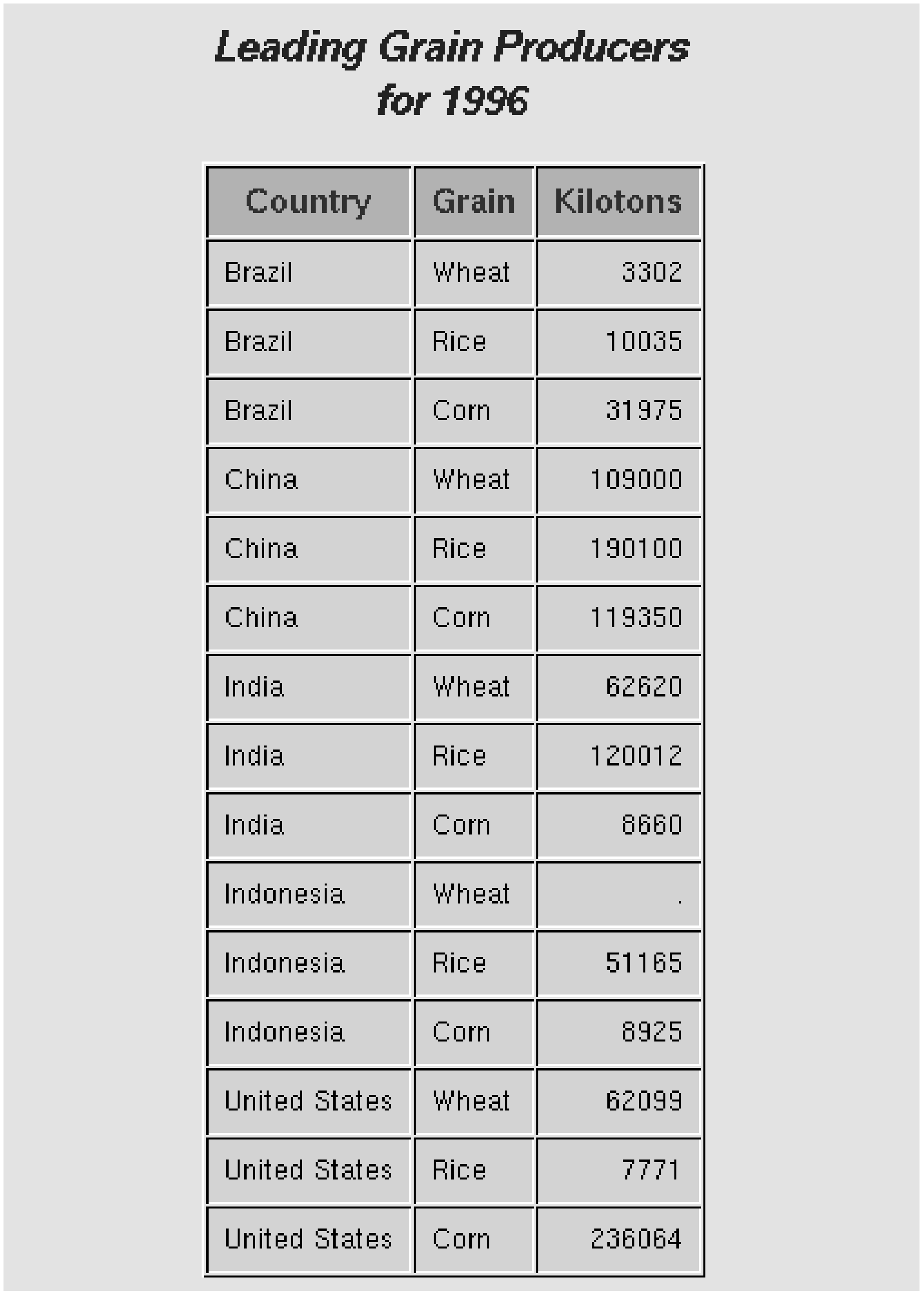
```
   /* The OPTIONS statement controls several aspects of the  */
   /* Listing output. None of these options affects the HTML */
   /* output.                                                */
options pagesize=60 linesize=64 nodate pageno=1;

proc format;
   value $cntry 'BRZ'='Brazil'
                'CHN'='China'
                'IND'='India'
                'INS'='Indonesia'
                'USA'='United States';
   /* The ODS HTML statement opens the HTML destination and   */
   /* creates HTML output. It sends all output objects to the */
   /* external file selectvars-body.htm in the current        */
   /* directory.                                               */
ods html body='selectvars-body.htm';

title 'Leading Grain Producers';
title2 'for 1996';

   /* The DATA step does not create a data set. Instead, it   */
   /* creates a data component and, eventually, an output     */
   /* object.                                                  */
data _null_;
   length Country $ 3 Type $ 5;
   input Year country $ type $ Kilotons;
   if year=1996;
   format country $cntry.;
   label type='Grain';

      /* The combination of the fileref PRINT and the ODS option */
      /* in the FILE statement routes the results of the DATA    */
      /* step to ODS. Two ODS destinations, the Listing and the  */
      /* HTML destinations, are open. Because no table definition*/
      /* is specified, ODS uses the default DATA step definition.*/
      /* The VARIABLES= suboption specifies the three variables  */
      /* that appear in the output.                               */
   file print ods=(variables=(country
                              type
                              kilotons));

      /* The _ODS_ option in the PUT statement writes all the */
      /* variables to its buffer. Then the PUT statement      */
      /* writes to the data component.                        */
   put _ods_;
   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
;

   /* 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.                             */
ods html close;
```

Output 16.3 Listing Output Produced by the Listing Destination

```
                     Leading Grain Producers                    1
                             for 1996
                 Country       Grain      Kilotons

              Brazil           Wheat           3302
              Brazil           Rice           10035
              Brazil           Corn           31975
              China            Wheat         109000
              China            Rice          190100
              China            Corn          119350
              India            Wheat          62620
              India            Rice          120012
              India            Corn            8660
              Indonesia        Wheat              .
              Indonesia        Rice           51165
              Indonesia        Corn            8925
              United States    Wheat          62099
              United States    Rice            7771
              United States    Corn          236064
```

Display 16.1 Body File Produced by the HTML Destination



Leading Grain Producers for 1996

Country	Grain	Kilotons
Brazil	Wheat	3302
Brazil	Rice	10035
Brazil	Corn	31975
China	Wheat	109000
China	Rice	190100
China	Corn	119350
India	Wheat	62620
India	Rice	120012
India	Corn	8660
Indonesia	Wheat	.
Indonesia	Rice	51165
Indonesia	Corn	8925
United States	Wheat	62099
United States	Rice	7771
United States	Corn	236064

Example 3: Specifying Attributes for a Column

This example assigns a label to the output object that it creates. It also specifies a label and a format for individual columns. The output is routed to three ODS destinations: the Listing destination, which is open by default, the HTML destination, which is opened by the ODS HTML statement, and the Printer destination, which is opened by the ODS PRINTER statement.

Operating Environment Information: 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 the SAS documentation for your operating environment. △

```
     /* The OPTIONS statement controls several aspects of the  */
     /* Listing output. NODATE and PAGENO= also affect the     */
     /* Printer output. None of these options affects the HTML */
     /* output.                                                */
  options pagesize=60 linesize=64 nodate pageno=1;

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

     /* The ODS HTML statement opens the HTML destination and  */
     /* creates HTML output. Subsequent output objects go to   */
     /* the body file. CONTENTS= and FRAME= create a frame file*/
     /* that includes a table of contents that links to the    */
     /* contents of the body file. The body file also appears  */
     /* in the frame. The ODS PRINTER statement opens the      */
     /* Printer destination and creates Printer output. It     */
     /* sends all output objects to the external file          */
     /* attribs.ps in the current directory.                   */
  ods html body='attribs-body.htm'
           contents='attribs-contents.htm'
           frame='attribs-frame.htm';
  ods printer file='attribs.ps';

  title 'Leading Grain Producers';
  title2 'for 1996';

     /* The DATA step does not create a data set. Instead, it   */
     /* creates a data component and, eventually, an output     */
     /* object.                                                 */
  data _null_;
     length Country $ 3 Type $ 5;
     input Year country $ type $ Kilotons;
     if year=1996;
     format country $cntry.;
     label type='Grain';

        /* The combination of the fileref PRINT and the ODS     */
        /* option in the FILE statement routes the results of   */
        /* the DATA step to ODS. Three ODS destinations--the    */
        /* Listing,the HTML, and the Printer destinations--are  */
        /* open. Because no table definition is specified, ODS   */
        /* uses the default DATA step definition.               */
     file print ods=(objectlabel='1996 Grain Production'

        /* The VARIABLES= suboption specifies the three variables */
        /* that appear in the output.                             */

                     variables=(country
                                type(label='Type of Grain')
                                kilotons(format=comma12.))
                                );

         /* The _ODS_ option in the PUT statement writes all    */
         /* the variables to its buffer. Then the PUT statement */
         /* writes to the data component.                        */
      put _ods_;
      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
   ;

      /* 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;
```

In the following Listing output, the label that is supplied as an attribute for the variable Grain becomes its column header. The format for Kilotons was supplied as an attribute in the ODS= option in the FILE statement.

Output 16.4 Listing Output Produced by the Listing Destination

```
                 Leading Grain Producers                  1
                         for 1996
                           Type
                            of
             Country       Grain      Kilotons

          Brazil           Wheat          3,302
          Brazil           Rice          10,035
          Brazil           Corn          31,975
          China            Wheat        109,000
          China            Rice         190,100
          China            Corn         119,350
          India            Wheat         62,620
          India            Rice         120,012
          India            Corn           8,660
          Indonesia        Wheat              .
          Indonesia        Rice          51,165
          Indonesia        Corn           8,925
          United States    Wheat         62,099
          United States    Rice           7,771
          United States    Corn         236,064
```

In the following HTML output, the object's label, which was supplied by OBJECTLABEL=, appears in the table of contents as the link to the output object. In the body file, the label that is supplied as an attribute for the variable Grain becomes its column header. The format for Kilotons was supplied as an attribute in the ODS= option in the FILE statement.

Display 16.2 Frame File Produced by the HTML Destination

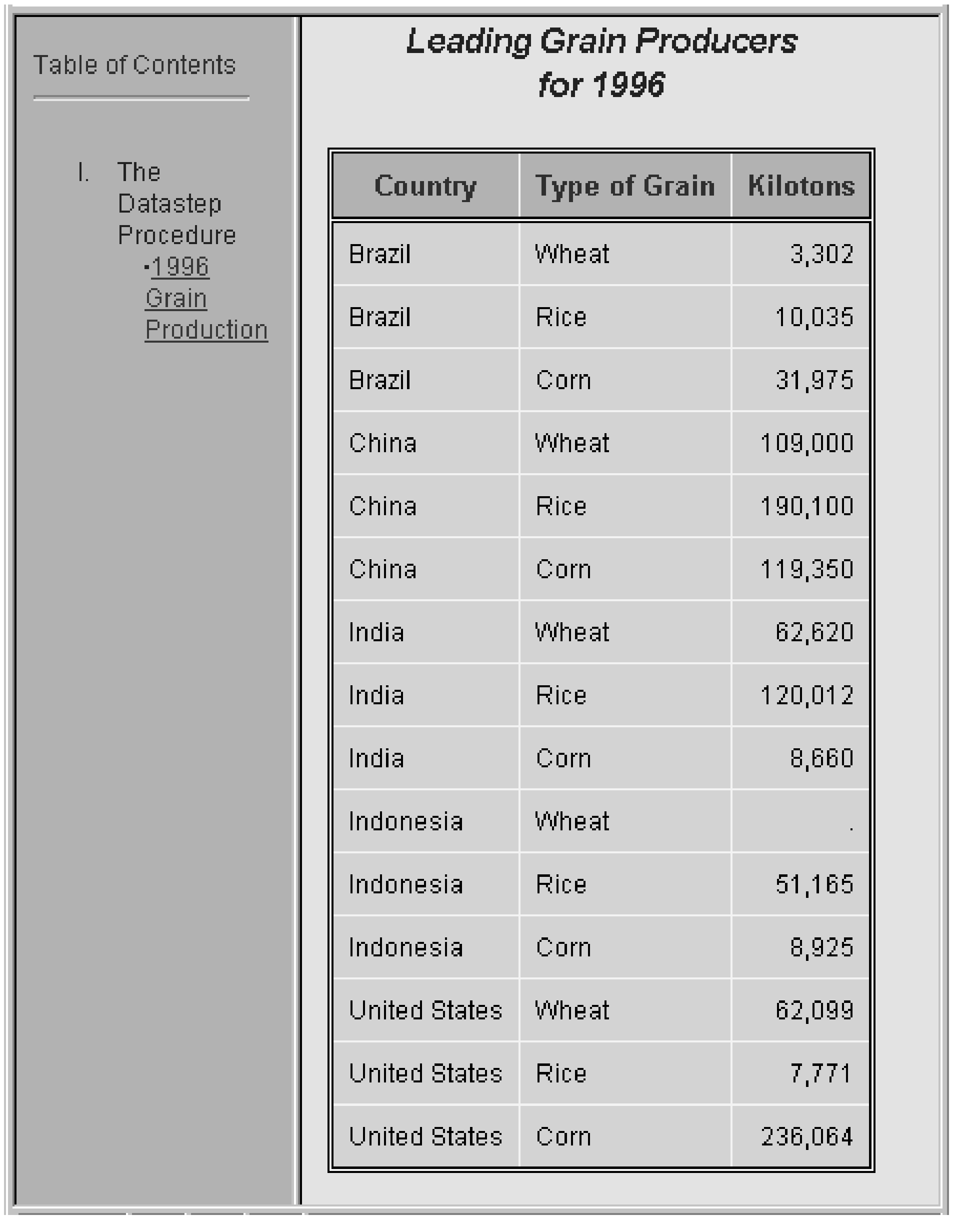


Table of Contents

I. The Datastep Procedure
- 1996 Grain Production

Leading Grain Producers for 1996

Country	Type of Grain	Kilotons
Brazil	Wheat	3,302
Brazil	Rice	10,035
Brazil	Corn	31,975
China	Wheat	109,000
China	Rice	190,100
China	Corn	119,350
India	Wheat	62,620
India	Rice	120,012
India	Corn	8,660
Indonesia	Wheat	.
Indonesia	Rice	51,165
Indonesia	Corn	8,925
United States	Wheat	62,099
United States	Rice	7,771
United States	Corn	236,064

In the following Printer output, the label that is supplied as an attribute for the variable Grain becomes its column header. The format for Kilotons was supplied as an attribute in the ODS= option in the FILE statement.

Display 16.3 Output from the Printer Destination

Leading Grain Producers for 1996 **1**

Country	Type of Grain	Kilotons
Brazil	Wheat	3,302
Brazil	Rice	10,035
Brazil	Corn	31,975
China	Wheat	109,000
China	Rice	190,100
China	Corn	119,350
India	Wheat	62,620
India	Rice	120,012
India	Corn	8,660
Indonesia	Wheat	.
Indonesia	Rice	51,165
Indonesia	Corn	8,925
United States	Wheat	62,099
United States	Rice	7,771
United States	Corn	236,064

Example 4: Building a Custom Table Definition for the TopN Report

This example uses PROC MEANS to create the SAS data set CHARITY, and PROC TEMPLATE to create a table for the output. The output is routed to two ODS destinations: the Listing destination, which is open by default, and the HTML destination, which is opened by the ODS HTML statement.

Operating Environment Information: 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 the SAS documentation for your operating environment. △

```
   /* The OPTIONS statement controls several aspects of the  */
   /* Listing output. None of these options affect the HTML */
   /* output.                                                */
options nodate pageno=1 pagesize=60 linesize=72;

proc format;
   value yrFmt . = " All";
   value $schFmt ' ' = "All    ";
run;
```

```
   /* The data set Charity contains information about high    */
   /* school students' volunteer work for charity. The        */
   /* variables give the name of the high school, the year of */
   /* the fundraiser, the first name of each student, the     */
   /* amount of money that each student raised, and the number*/
   /* of hours that each student volunteered.                 */
data Charity;
   input School $ 1-7 Year 9-12 Name $ 14-20 moneyRaised 22-26
         hoursVolunteered 28-29;
   format moneyRaised dollar8.2;
   format hoursVolunteered f3.0;
   format Year yrFmt.;
   format School schFmt.;
   label School = "Schools";
   label Year = "Years";
   retain yearmin yearmax;
   yearmin=min(yearmin,year);
   yearmax=max(yearmax,year);
   call symput('first_year',put(yearmin,4.));
   call symput('last_year', put(yearmax,4.));
   datalines;
Monroe  1992 Allison 31.65 19
Monroe  1992 Barry   23.76 16
Monroe  1992 Candace 21.11  5

… more data lines …

Kennedy 1994 Sid     27.45 25
Kennedy 1994 Will    28.88 21
Kennedy 1994 Morty   34.44 25
;

     /* This PROC MEANS step analyzes the data for the one way  */
     /* combination of the class variables and across all       */
     /* observations. It creates an output data set that        */
     /* includes variables for the total and average amount of  */
     /* money raised. The data set also includes new variables  */
     /* for the top three amounts of money raised, the names of */
     /* the three students who raised the money, the years when */
     /* the students raised the money, and the schools that the */
     /* students attended. For a detailed explanation of the    */
     /* MEANS procedure, see the SAS Procedures Guide.          */
proc means data=Charity descendTypes charType noprint;
   class School Year;
   var moneyRaised;
   types () School year;
   output out=top3list sum= mean=
      idgroup ( max(moneyRaised) out[3](moneyRaised name school year)= )
      / autoname;
run;

   /* The PROC PRINT step generates traditional Listing      */
   /* output of the output data set that PROC MEANS created. */
```

```
proc print data=top3list noobs;
   title 'Fund Raising Results';
run;

title;

   /* The ODS HTML statement opens the HTML destination and    */
   /* creates HTML output. It sends all output objects to the  */
   /* external file topn-body.htm in the current directory.    */
   /* Some browsers require an extension of .htm or .html on   */
   /* the filename.                                            */
 ods html body='topn-body.htm';

   /* The DEFINE statement creates the table definition        */
   /* means.topn in the first template store in the path that  */
   /* is available to write to. BY default, this template      */
   /* store is SASUSER.TEMPLAT.                                */
proc template;
   define table means.topn;
   mvar first_year last_year sysdate9;

      /* The COLUMN statement declares these symbols as columns */
      /* in the table and specifies their order in the table.   */
      /* If a column name appears in parentheses, PROC TEMPLATE */
      /* stacks the values of all variables that use that column*/
      /* definition one below the other in the output object.   */
      /* Variables are assigned a column definition in the DATA */
      /* step that appears later in the program.                */
   column class sum mean (raised) (name) (school) (year);

      /* These three table attributes affect the presentation   */
      /* of the output object in the Listing output. They have  */
      /* no effect on its presentation in the HTML output.      */
      /* DOUBLE_SPACE= double spaces the rows of the output     */
      /* object. OVERLINE= and UNDERLINE= draw a continuous     */
      /* line before the first row of the table and after the   */
      /* last row of the table.                                 */
   double_space=on;
   overline=on;
   underline=on;

      /* The HEADER statement declares table_header_1 and       */
      /* table_header_2 as headers in the table and specifies   */
      /* the order in which the headers appear in the output    */
      /* object.                                                */
   header table_header_1 table_header_2;

      /* The  DEFINE statement and its substatement and         */
      /* attribute define table_header_1. The TEXT statement    */
      /* specifies the text of the header. The STYLE= attribute */
      /* alters the style element that renders the table header.*/
      /* The END statement ends the header definition.          */
   define table_header_1;
      text "Top Three Fund Raisers";
   style=header{font_size=6};
end;

   /* The  DEFINE statement and its substatement and         */
   /* attribute define table_header_2. The TEXT statement    */
   /* uses text and the macro variables FIRST_YEAR and       */
   /* LAST_YEAR to specify the contents of the header.       */
   /* The END statement ends the header definition.          */
define table_header_2;
   text "from " first_year " to " last_year;
   space=1;
end;

   /* The  DEFINE statement and its substatement and         */
   /* attribute define table_footer. The FOOTER argument     */
   /* declares table_footer as a footer.                     */
define footer table_footer;
   text "(report generated on " sysdate9 ")";
   split="*";
   style=header{font_size=2};
end;

   /* The DEFINE statement and its attributes create the    */
   /* column definition class. (The COLUMN statement        */
   /* earlier in the program declared class as a column.)   */
define class;
   generic=on;
   id=on;
   vjust=top;
   style=data;
end;

   /* Each of these DEFINE statements and its attributes    */
   /* creates a column definition. The END statement ends   */
   /* the definition.                                       */
define sum;
   generic=on;
   header="Total Dollars Raised";
   vjust=top;
end;

define mean;
   generic=on;
   header="Average Dollars per Student";
   vjust=top;
end;

define raised;
   generic=on;
   header="Individual Dollars";
end;

define name;
   generic=on;
```

```
      header="Student";
   end;

   define school;
      generic=on;
      header="School";
   end;

   define year;
      generic=on;
      header="Year";
   end;

      /* This END statement ends the table definition. The RUN */
      /* statement ends the PROC TEMPLATE step.                */
   end;
run;

   /* This DATA step does not create a data set. Instead, it   */
   /* creates a data component and, eventually, an output      */
   /* object. The SET statement reads the data set TOP3LIST,   */
   /* which PROC MEANS created.                                */
data _null_;
   set top3list;

      /* The combination of the fileref PRINT and the ODS option */
      /* in the FILE statement routes the results of the DATA    */
      /* step to ODS. The TEMPLATE= suboption tells ODS to use   */
      /* the table definition named means.topn, which was just   */
      /* created with PROC TEMPLATE.                             */
   file print ods = (
      template='means.topn'

      /* The COLUMNS= suboption places DATA step variables into  */
      /* columns that are defined in the table definition.       */
      columns=(
         class=school(generic=on)
         class=year(generic=on)
         sum=moneyRaised_sum(generic=on)
         mean=moneyRaised_mean(generic=on)
         raised=moneyRaised_1(generic=on)
         raised=moneyRaised_2(generic=on)
         raised=moneyRaised_3(generic=on)
         name=name_1(generic=on)
         name=name_2(generic=on)
         name=name_3(generic=on)
         school=school_1(generic=on)
         school=school_2(generic=on)
         school=school_3(generic=on)
         year=year_1(generic=on)
         year=year_2(generic=on)
         year=year_3(generic=on)
         )
      );
   /* The _ODS_ option and the PUT statement write the data values */
   /* for all columns to the data component.                       */
put _ods_;
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.   */
ods html close;
```

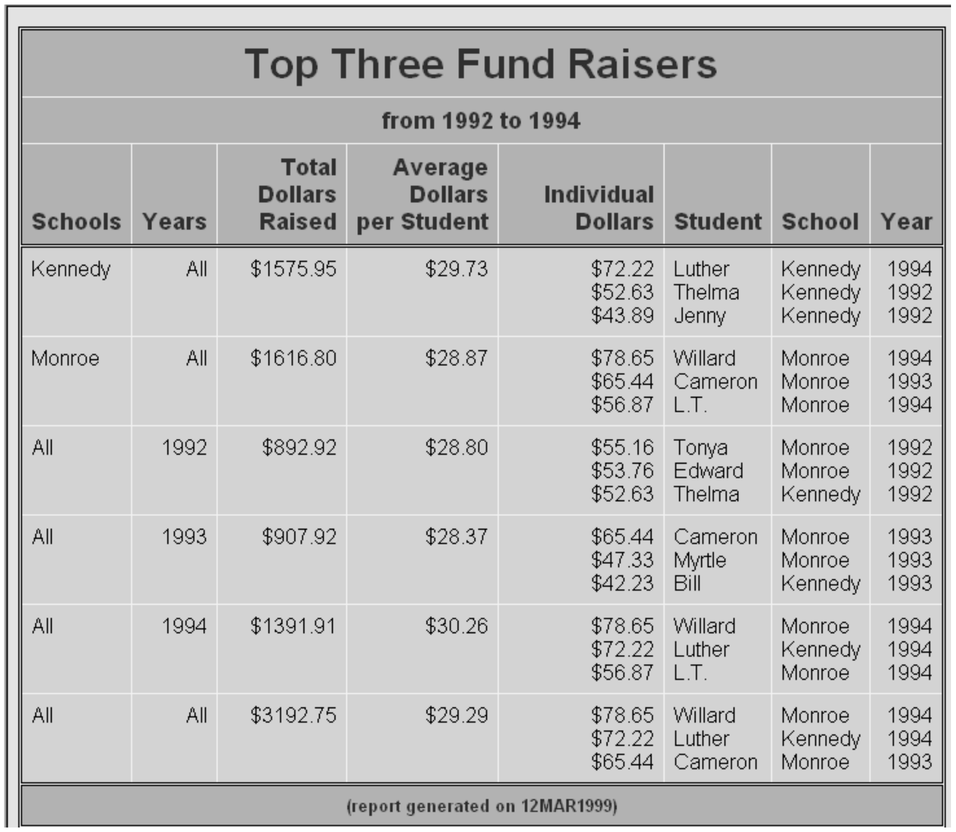
Output 16.5 PROC PRINT Listing Output for the Output Data Set from PROC MEANS

```
                           Fund Raising Results                         1

                               money    money
                             Raised_  Raised_  money    money    money
School  Year _TYPE_ _FREQ_      Sum     Mean Raised_1 Raised_2 Raised_3

Kennedy  All   10      53  $1575.95   $29.73   $72.22   $52.63   $43.89
Monroe   All   10      56  $1616.80   $28.87   $78.65   $65.44   $56.87
All     1992   01      31   $892.92   $28.80   $55.16   $53.76   $52.63
All     1993   01      32   $907.92   $28.37   $65.44   $47.33   $42.23
All     1994   01      46  $1391.91   $30.26   $78.65   $72.22   $56.87
All      All   00     109  $3192.75   $29.29   $78.65   $72.22   $65.44



Name_1  Name_2  Name_3  School_1 School_2 School_3 Year_1 Year_2 Year_3

Luther  Thelma  Jenny   Kennedy  Kennedy  Kennedy   1994   1992   1992
Willard Cameron L.T.    Monroe   Monroe   Monroe    1994   1993   1994
Tonya   Edward  Thelma  Monroe   Monroe   Kennedy   1992   1992   1992
Cameron Myrtle  Bill    Monroe   Monroe   Kennedy   1993   1993   1993
Willard Luther  L.T.    Monroe   Kennedy  Monroe    1994   1994   1994
Willard Luther  Cameron Monroe   Kennedy  Monroe    1994   1994   1993
```

Output 16.6 Using a Customized Table to Produce Listing Output

```
                                                                          2
                         Top Three Fund Raisers
                           from 1992 to 1994

                                Average
                         Total  Dollars
                       Dollars      per  Individual
Schools  Years          Raised  Student     Dollars  Student  School   Year
----------------------------------------------------------------------
Kennedy    All  $1575.95   $29.73      $72.22  Luther   Kennedy  1994
                                           $52.63  Thelma   Kennedy  1992
                                           $43.89  Jenny    Kennedy  1992

Monroe     All  $1616.80   $28.87      $78.65  Willard  Monroe   1994
                                           $65.44  Cameron  Monroe   1993
                                           $56.87  L.T.     Monroe   1994

All       1992   $892.92   $28.80      $55.16  Tonya    Monroe   1992
                                           $53.76  Edward   Monroe   1992
                                           $52.63  Thelma   Kennedy  1992

All       1993   $907.92   $28.37      $65.44  Cameron  Monroe   1993
                                           $47.33  Myrtle   Monroe   1993
                                           $42.23  Bill     Kennedy  1993

All       1994  $1391.91   $30.26      $78.65  Willard  Monroe   1994
                                           $72.22  Luther   Kennedy  1994
                                           $56.87  L.T.     Monroe   1994

All        All  $3192.75   $29.29      $78.65  Willard  Monroe   1994
                                           $72.22  Luther   Kennedy  1994
                                           $65.44  Cameron  Monroe   1993
----------------------------------------------------------------------
                   (report generated on 20JUL1999)
```

Display 16.4 HTML Output for the TopN Report



Top Three Fund Raisers

from 1992 to 1994

Schools	Years	Total Dollars Raised	Average Dollars per Student	Individual Dollars	Student	School	Year
Kennedy	All	$1575.95	$29.73	$72.22 $52.63 $43.89	Luther Thelma Jenny	Kennedy Kennedy Kennedy	1994 1992 1992
Monroe	All	$1616.80	$28.87	$78.65 $65.44 $56.87	Willard Cameron L.T.	Monroe Monroe Monroe	1994 1993 1994
All	1992	$892.92	$28.80	$55.16 $53.76 $52.63	Tonya Edward Thelma	Monroe Monroe Kennedy	1992 1992 1992
All	1993	$907.92	$28.37	$65.44 $47.33 $42.23	Cameron Myrtle Bill	Monroe Monroe Kennedy	1993 1993 1993
All	1994	$1391.91	$30.26	$78.65 $72.22 $56.87	Willard Luther L.T.	Monroe Kennedy Monroe	1994 1994 1994
All	All	$3192.75	$29.29	$78.65 $72.22 $65.44	Willard Luther Cameron	Monroe Kennedy Monroe	1994 1994 1993

(report generated on 12MAR1999)

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

SAS Language Reference: Concepts

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1–58025–441–1

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

1st printing, November 1999