SAS Certification Prep Guide: Base Programming for SAS 9, Third Edition

by SAS Institute

SAS Institute. (c) 2011. Copying Prohibited.

Reprinted for Shane Mc Carthy, Accenture

shane.mc.carthy@accenture.com

Reprinted with permission as a subscription benefit of **Skillport**,
http://skillport.books24x7.com/

All rights reserved. Reproduction and/or distribution in whole or in part in electronic,paper or other forms without written permission is prohibited.

Chapter 1: Base Programming

Overview

Introduction

To program effectively using SAS, you need to understand basic concepts about SAS programs and the SAS files that they process. In particular, you need to be familiar with SAS data sets.

In this chapter, you'll examine a simple SAS program and see how it works. You'll learn details about SAS data sets (which are files that contain data that is logically arranged in a form that SAS can understand). You'll see how SAS data sets are stored temporarily or permanently in SAS libraries. Finally, you'll learn how to use SAS windows to manage your SAS session and to process SAS programs.

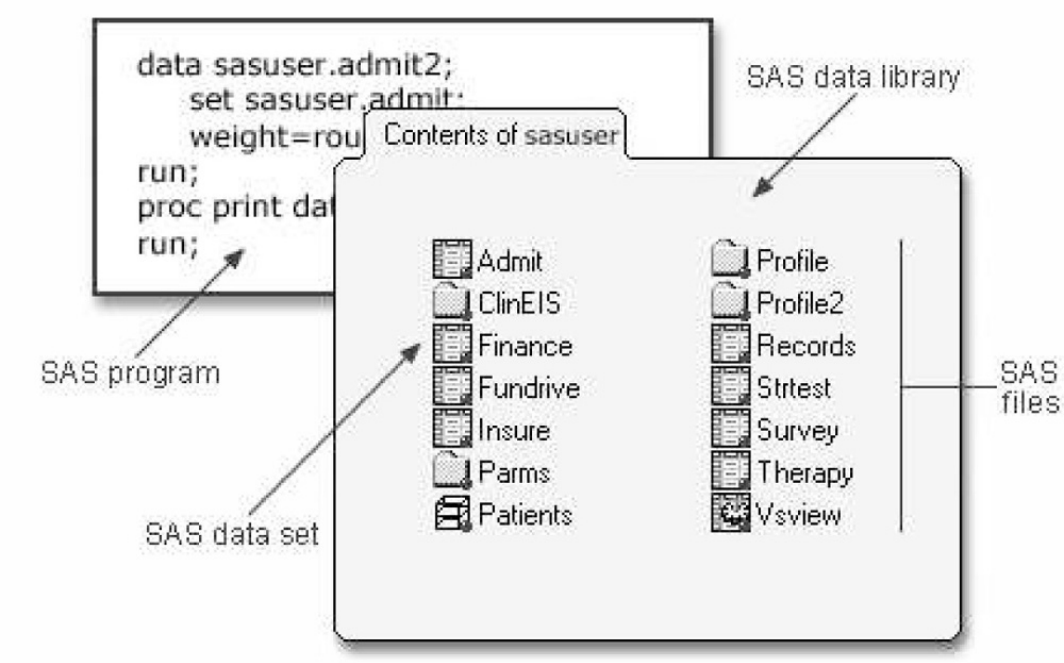


Figure 1.1: SAS Library with SAS Data Sets and Data Files

Objectives

In this chapter, you learn about

- the structure and components of SAS programs
- the steps involved in processing SAS programs
- SAS libraries and the types of SAS files that they contain
- temporary and permanent SAS libraries
- the structure and components of SAS data sets
- the SAS windowing environment.

SAS Programs

You can use SAS programs to access, manage, analyze, or present your data. Let's begin by looking at a simple SAS program.

A Simple SAS Program

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

This program uses an existing SAS data set to create a new SAS data set containing a subset of the original data set. It then prints a listing of the new data set using PROC PRINT. A SAS data set is a data file that is formatted in a way that SAS can understand.

```
data sasuser.admit2;
   set sasuser.admit;
   where age>39;
run;
proc print data=sasuser.admit2;
run;
```

Let's see how this program works.

Components of SAS Programs

The sample SAS program contains two steps: a DATA step and a PROC step.

```
data sasuser.admit2;
   set sasuser.admit;
   where age>39;
run;
proc print data=sasuser.admit2;
run;
```

These two types of steps, alone or combined, form most SAS programs.

A SAS program can consist of a DATA step or a PROC step or any combination of DATA and PROC steps.

Figure 1.2: Components of a SAS Program

DATA steps typically create or modify SAS data sets. They can also be used to produce custom-designed reports. For example, you can use DATA steps to

- put your data into a SAS data set
- compute values
- check for and correct errors in your data

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

- produce new SAS data sets by subsetting, supersetting, merging, and updating existing data sets.

In the previous example, the DATA step produced a new SAS data set containing a subset of the original data set. The new data set contains only those observations with an age value greater than 39.

PROC (procedure) steps invoke or call pre-written routines that enable you to analyze and process the data in a SAS data set. PROC steps typically present the data in the form of a report. They sometimes create new SAS data sets that contain the results of the procedure. PROC steps can list, sort, and summarize data. For example, you can use PROC steps to

- create a report that lists the data
- produce descriptive statistics
- create a summary report
- produce plots and charts.

Characteristics of SAS Programs

Next let's look at the individual statements in our sample program. SAS programs consist of SAS statements. A SAS statement has two important characteristics:

- It usually begins with a SAS keyword.
- It always ends with a semicolon.

As you've seen, a DATA step begins with a DATA statement, which begins with the keyword DATA. A PROC step begins with a PROC statement, which begins with the keyword PROC. Our sample program contains the following statements:

Table 1.1: SAS Program Statements

Statements	Sample Program Code
a DATA statement	`data sasuser.admit2;`
a SET statement	`   set sasuser.admit;`
Additional programming statements	`   where age>39;`
a RUN statement	`run;`
a PROC PRINT statement	`proc print data=sasuser.admit2;`
another RUN statement	`run;`

Layout for SAS Programs

SAS statements are free-format. This means that

- they can begin and end anywhere on a line
- one statement can continue over several lines
- several statements can be on the same line.

Blanks or special characters separate *words* in a SAS statement.

Additional Note You can specify SAS statements in uppercase or lowercase. In most situations, text that is enclosed in quotation marks is case sensitive.

You've examined the general structure of our sample program. But what happens when you run the program?

Processing SAS Programs

When you submit a SAS program, SAS begins reading the statements and checking them for errors.

DATA and PROC statements signal the beginning of a new step. The RUN statement (for most procedures and the DATA

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

step) and the QUIT statement (for some procedures) mark step boundaries. The beginning of a new step (DATA or PROC) also implies the end of the previous step. At a step boundary, SAS executes any statements that have not previously executed and ends the step. In our sample program, each step ends with a RUN statement.

```
data sasuser.admit2;
   set sasuser.admit;
   where age>39;
run;
proc print data=sasuser.admit2;
run;
```

Additional Note Though the RUN statement is not always required between steps in a SAS program, using it can make the SAS program easier to read and debug, and it makes the SAS log easier to read.

Log Messages

Each time a step is executed, SAS generates a log of the processing activities and the results of the processing. The SAS log collects messages about the processing of SAS programs and about any errors that occur.

When SAS processes our sample program, you see the log messages shown below. Notice that you get separate sets of messages for each step in the program.

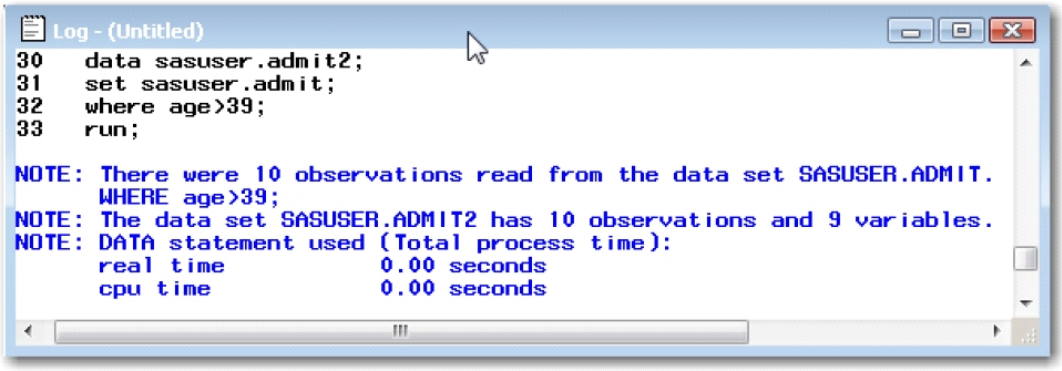


Figure 1.3: Log Messages

Results of Processing

- DATA step Output

  Suppose you submit the sample program below:

  ```
  data sasuser.admit2;
  set sasuser.admit;
     where age>39;
  run;
  ```

  When the program is processed, it creates a new SAS data set (sasuser.admit2) containing only those observations with age values greater than 39. The DATA step creates a new data set and produces messages in the SAS log, but it does not create a report or other output.

- Procedure Output

  If you add a PROC PRINT statement to this same example, the program produces the same new data set as before, but it also creates the following report, which is displayed in HTML:

  ```
  data sasuser.admit2;
  set sasuser.admit;
     where age>39;
  run;
  proc print data=sasuser.admit2;
  run;
  ```

The SAS System

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2523	Johnson, R	F	43	31	63	137	MOD	149.75
2	2539	LaMance, K	M	51	4	71	158	LOW	124.80
3	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80
4	2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75
5	2575	Quigley, M	F	40	8	69	163	HIGH	124.80
6	2576	Cameron, L	M	47	5	72	173	MOD	124.80
7	2579	Underwood, K	M	60	22	71	191	LOW	149.75
8	2584	Takahashi, Y	F	43	29	65	123	MOD	124.80
9	2589	Wilcox, E	F	41	16	67	141	HIGH	149.75
10	2595	Warren, C	M	54	7	71	183	MOD	149.75

Figure 1.4: PRINT Procedure Output

Additional Note Throughout this book, procedure output is shown in HTML in the style shown above unless otherwise noted.

You've seen the results of submitting our sample program. For other SAS programs, the results of processing might vary:

- Other Types of Procedural Output
  - Some SAS programs open an interactive window (a window that you can use to directly modify data), such as the REPORT window.

```
proc report data=sasuser.admit;
   columns id name sex age actlevel;
   run;
```

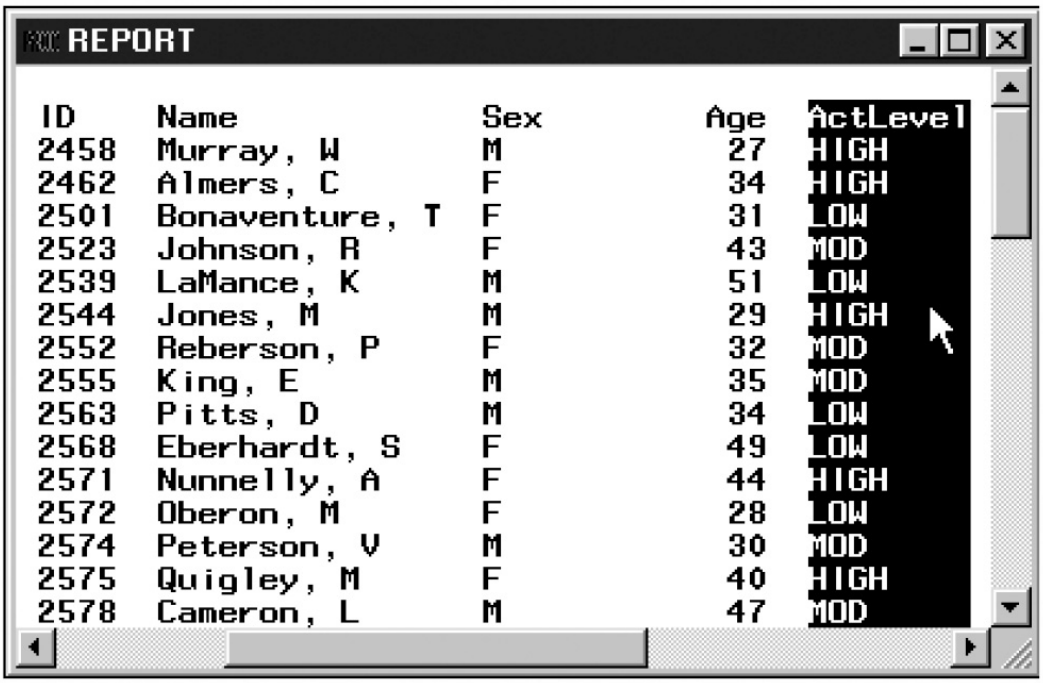


Figure 1.5: Interactive Report Window

  - SAS programs often invoke procedures that create output in the form of a report, as is the case with the TABULATE procedure:

```
proc tabulate data=sasuser.admit;
   class sex;
   var height weight;
```


Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

```
   table sex*(height weight),mean;
run;
```

The SAS System

		Mean
Sex		
F	Height	64.82
	Weight	141.73
M	Height	72.00
	Weight	172.80

Figure 1.6: TABULATE Procedure Output

- Other SAS programs perform tasks such as sorting and managing data, which have no visible results except for messages in the log. (All SAS programs produce log messages, but some SAS programs produce only log messages.)

```
proc copy in=sasuser out=work;
   select admit;
run;
```

```
36   proc copy in=sasuser out=work;
37      select admit;
38   run;

NOTE: Copying SASUSER.ADMIT to WORK.ADMIT (memtype=DATA).
NOTE: There were 21 observations read from the data set SASUSER.ADMIT.
NOTE: The data set WORK.ADMIT has 21 observations and 9 variables.
NOTE: PROCEDURE COPY used (Total process time):
      real time           0.15 seconds
      cpu time            0.04 seconds
```

Figure 1.7: Log Output

SAS Libraries

So far you've learned about SAS programs. Now let's look at SAS libraries to see how SAS data sets and other SAS files are organized and stored.

How SAS Files Are Stored

Every SAS file is stored in a SAS library, which is a collection of SAS files. A SAS data library is the highest level of organization for information within SAS.

For example, in the Windows and UNIX environments, a library is typically a group of SAS files in the same folder or directory.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

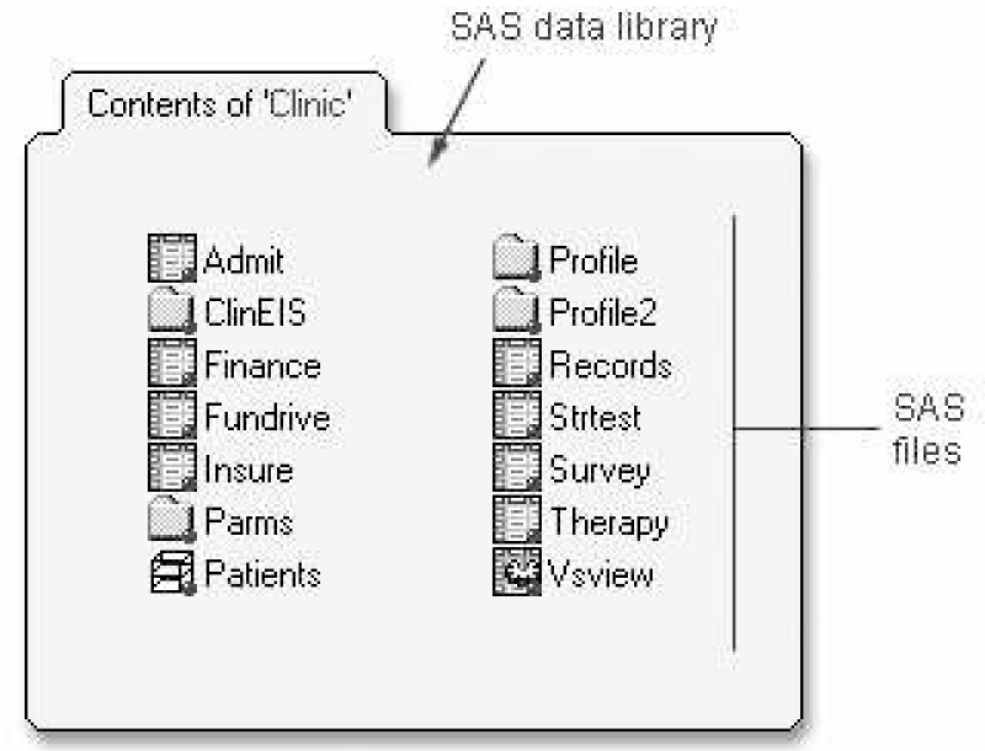


Figure 1.8: SAS Data Library

The table below summarizes the implementation of SAS libraries in various operating environments.

Table 1.2: Environments and SAS Libraries

In this environment…	A SAS library is…
Windows, UNIX, OpenVMS, OS/2 (directory based-systems)	a group of SAS files that are stored in the same directory. Other files can be stored in the directory, but only the files that have SAS file extensions are recognized as part of the SAS library. **Additional Note** For more information, see the SAS documentation for your operating environment.
z/OS (OS/390)	a specially formatted host data set in which only SAS files are stored.

Storing Files Temporarily or Permanently

Depending on the library name that you use when you create a file, you can store SAS files temporarily or permanently.

Reprinted for OET7P/856094, Accenture
SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Figure 1.9: Temporary SAS Data Library

Figure 1.10: Permanent SAS Data Library

Table 1.3: Temporary and Permanent SAS Libraries

Temporary SAS libraries last only for the current SAS session.	Storing files temporarily: If you don't specify a library name when you create a file (or if you specify the library name Work), the file is stored in the temporary SAS data library. When you end the session, the temporary library and all of its files are deleted.
Permanent SAS Libraries are available to you during subsequent SAS sessions.	Storing files permanently: To store files permanently in a SAS data library, you specify a library name other than the default library name Work. For example, by specifying the library name sasuser when you create a file, you specify that the file is to be stored in a permanent SAS data library until you delete it.

Additional Note You can learn how to set up permanent SAS libraries in Chapter 2, "Referencing Files and Setting Options," on page 41.

Referencing SAS Files

Two-Level Names

To reference a permanent SAS data set in your SAS programs, you use a two-level name consisting of the library name and the filename, or data set name:

libref.filename

In the two-level name, *libref* is the name of the SAS data library that contains the file, and *filename* is the name of the file, or data set. A period separates the libref and filename.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

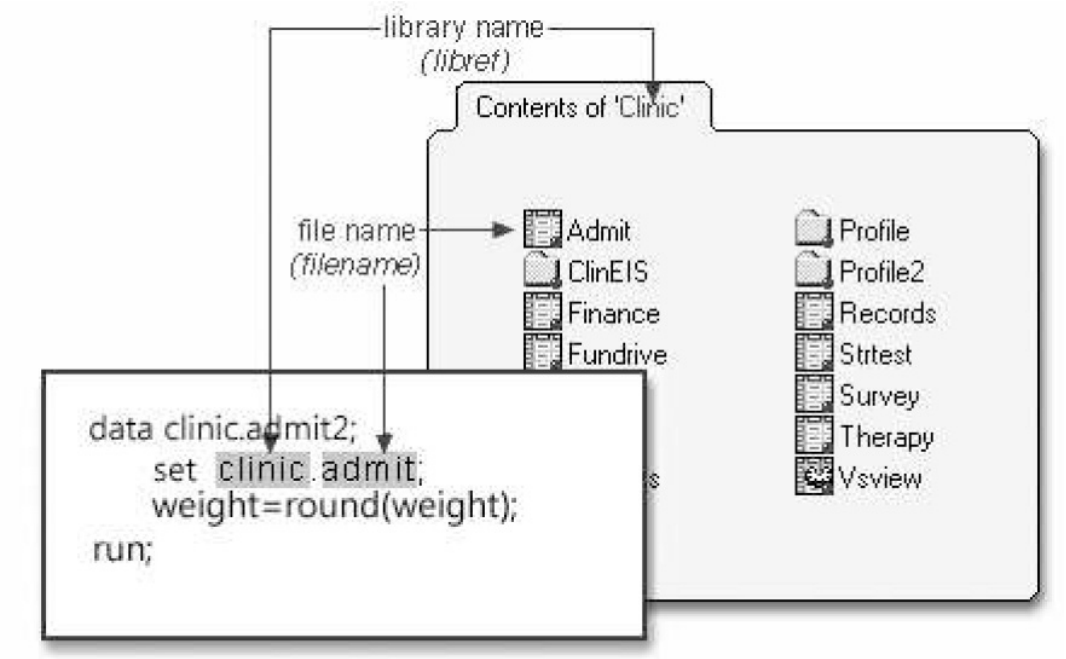


Figure 1.11: Two-Level SAS Name

For example, suppose we want to create a new permanent sas library named Clinic. In our sample program, Clinic.Admit is the two-level name for the SAS data set Admit, which is stored in the library named Clinic. Notice that the LIBNAME statement is used to define the libref, Clinic, and to give SAS the physical location of the data files.

```
libname clinic 'c:\Users\Name\sasuser';
data clinic.admit2;
   set clinic.admit;
   weight =round(weight);
run;
```

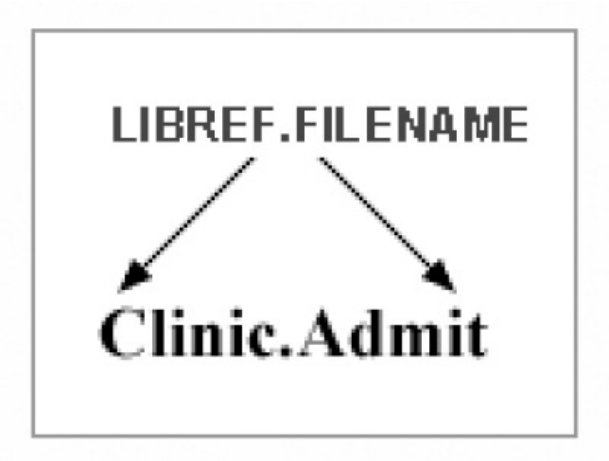


Figure 1.12: Two-Level Name Clinic.Admit

Referencing Temporary SAS Files

To reference temporary SAS files, you can specify the default libref Work, a period, and the filename. For example, the two-level name Work.Test references the SAS data set named *Test* that is stored in the temporary SAS library Work.

Reprinted for OET7P/856094, Accenture
SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

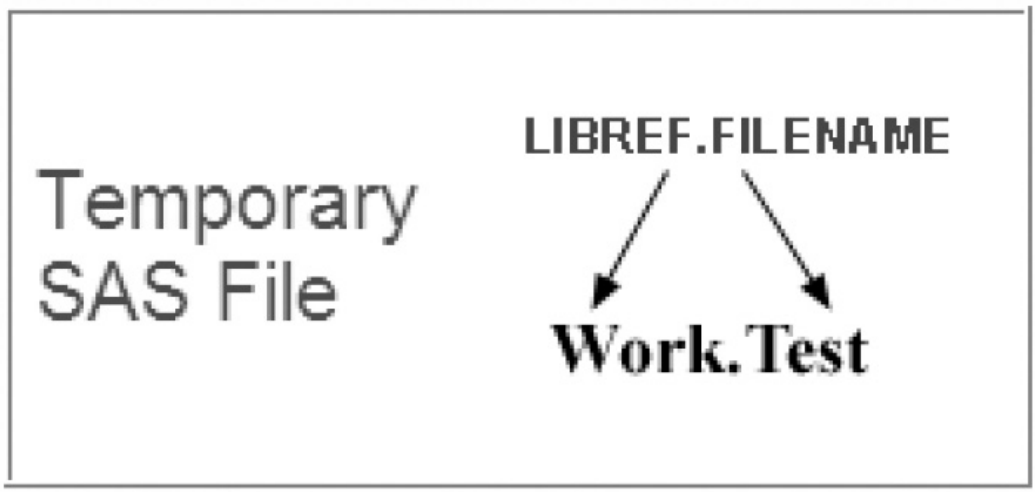


Figure 1.13: Two-Level Temporary SAS Library Work.Test

Alternatively, you can use a one-level name (the *filename* only) to reference a file in a temporary SAS library. When you specify a one-level name, the default libref Work is assumed. For example, the one-level name Test also references the SAS data set named Test that is stored in the temporary SAS library Work.

Figure 1.14: One-Level Temporary SAS Library Test

Referencing Permanent SAS Files

You can see that Clinic.Admit and Clinic.Admit2 are permanent SAS data sets because the library name is Clinic, not Work.

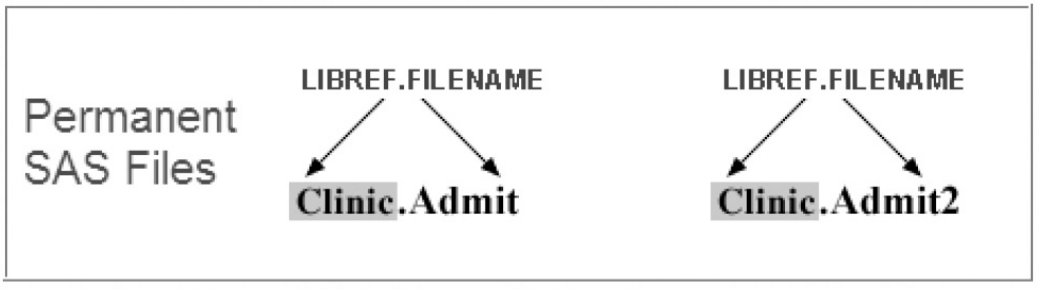


Figure 1.15: Referencing Permanent SAS Files

So, referencing a SAS file in any library except Work indicates that the SAS file is stored permanently. For example, when

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

our sample program creates Clinic.Admit2, it stores the new Admit2 data set permanently in the SAS library Clinic.

Rules for SAS Names

These rules apply only to the filename portion of a SAS data set name. A libref can have a length of only eight characters.

SAS data set names and variable names

- can be 1 to 32 characters long
- must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_)
- can continue with any combination of numbers, letters, or underscores.

These are examples of valid data set names and variable names:

- Payroll
- LABDATA1995_1997
- _EstimatedTaxPayments3

SAS Data Sets

So far, you've seen the components and characteristics of SAS programs, including how they reference SAS data sets. Data sets are one type of SAS file. There are other types of SAS files (such as catalogs), but this chapter focuses on SAS data sets. For most procedures, data must be in the form of a SAS data set to be processed. Now let's take a closer look at SAS data sets.

Overview of Data Sets

As you saw in our sample program, for many of the data processing tasks that you perform with SAS, you

- access data in the form of a SAS data set
- analyze, manage, or present the data.

Conceptually, a SAS data set is a file that consists of two parts: a descriptor portion and a data portion. Sometimes a SAS data set also points to one or more indexes, which enable SAS to locate rows in the data set more efficiently. (The data sets that you work with in this chapter do not contain indexes.)

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

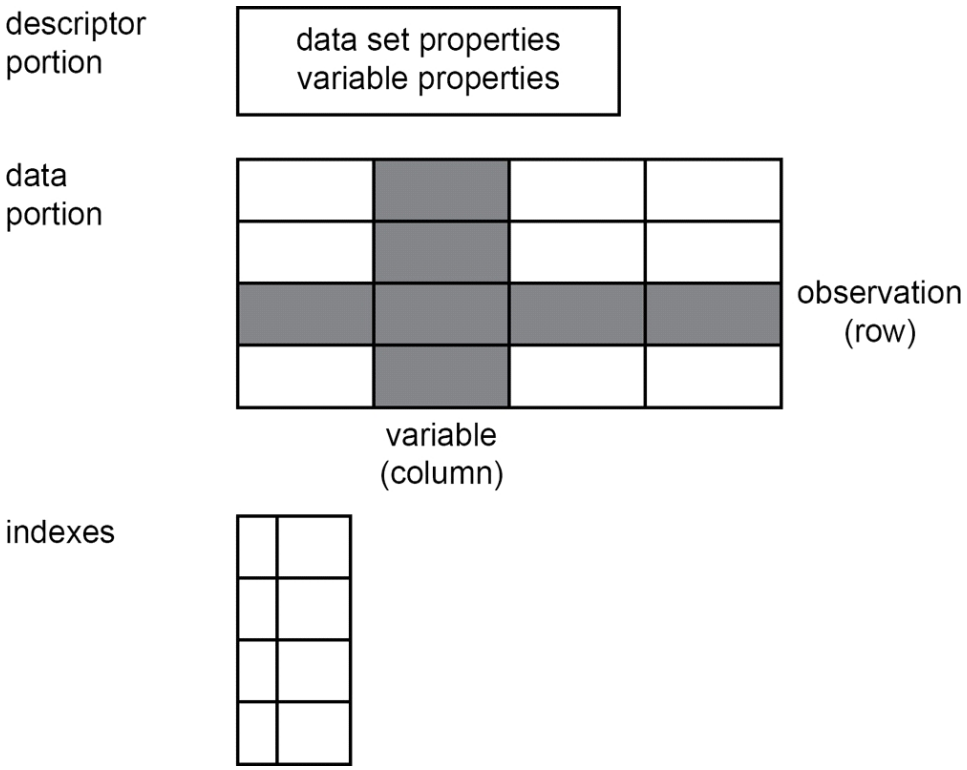


Figure 1.16: Parts of a SAS Data Set

Descriptor Portion

The descriptor portion of a SAS data set contains information about the data set, including

- the name of the data set
- the date and time that the data set was created
- the number of observations
- the number of variables.

Let's look at another SAS data set. The table below lists part of the descriptor portion of the data set sasuser.insure, which contains insurance information for patients who are admitted to a wellness clinic. (It's a good idea to give your data set a name that is descriptive of the contents.)

Table 1.4: Descriptor Portion of Attributes in a SAS Data Set sasuser.insure

Data Set Name:	**sasuser.INSURE**
Member Type:	DATA
Engine:	V9
Created:	10:05 Tuesday, February 16, 2011
Observations:	21
Variables:	7
Indexes:	0
Observation Length:	64

Data Portion

The data portion of a SAS data set is a collection of data values that are arranged in a rectangular table. In the example below, the name **`Murray`** is a data value, Policy **`32668`** is a data value, and so on.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

ID	Name	Policy	Company	PctInsured	Total	BalanceDue
24-58	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2601	Bon aventure. T	87795	A&R	80	47.38	9.48
2623	Johnson, R	39022	ACME	60	122.07	61.04

Figure 1.17: Parts of a SAS Data Set: Data: Data Portion

Observations (Rows)

Rows (called observations) in the data set are collections of data values that usually relate to a single object. The values **`2458, Murray, 32668, Mutuality, 100, 98.64`** and **`0.00`** comprise a single observation in the data set shown below.

Observation

ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	80	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

Figure 1.18: Parts of a SAS Data Set: Observations

This data set has seven observations, each containing information about an individual. A SAS data set can store any number of observations.

Variables (Columns)

Columns (called variables) in the data set are collections of values that describe a particular characteristic. The values **`2458, 2462, 2501`**, and **`2523`** comprise the variable ID in the data set shown below.

Variables

ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	80	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

Figure 1.19: Parts of a SAS Data Set: Variable

This data set contains seven variables: ID, Name, Policy, Company, PctInsured, Total, and BalanceDue. A SAS data set can store thousands of variables.

Missing Values

Every variable and observation in a SAS data set must have a value. If a data value is unknown for a particular observation, a missing value is recorded in the SAS data set.

Missing Value

ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	.	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Figure 1.20: Missing Data Values

Variable Attributes

In addition to general information about the data set, the descriptor portion contains information about the properties of each variable in the data set. The properties information includes the variable's name, type, length, format, informat, and label.

When you write SAS programs, it's important to understand the attributes of the variables that you use. For example, you might need to combine SAS data sets that contain same-named variables. In this case, the variables must be the same type (character or numeric).

The following is a partial listing of the attribute information in the descriptor portion of the SAS data set insure.policy. First, let's look at the name, type, and length variable attributes.

Table 1.5: Variable Attributes in the Descriptor Portion of a SAS Data Set insure.policy

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Name

Each variable has a name that conforms to SAS naming conventions. Variable names follow exactly the same rules as SAS data set names. Like data set names, variable names

- can be 1 to 32 characters long
- must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_)
- can continue with any combination of numbers, letters, or underscores.

Table 1.6: Variable Name Attributes

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Additional Note Your site may choose to restrict variable names to those valid in SAS 6, to uppercase variable names automatically, or to remove all restrictions on variable names.

Type

A variable's type is either character or numeric.

- Character variables, such as Name (shown below), can contain *any values*.
- Numeric variables, such as Total (shown below), can contain *only numeric values* (the numerals 0 through 9, +, -, ., and E for scientific notation).

Table 1.7: Type Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

A variable's type determines how missing values for a variable are displayed. In the following data set, Name and Sex are character variables, and Age and Weight are numeric variables.

- For character variables such as Name, a *blank* represents a missing value.
- For numeric variables such as Age, a *period* represents a missing value.

Name	Sex	Age	Weight
	M	48	128.6
Laverne	M	58	158.3
Jaffe	F	.	115.5
Wilson	M	28	170.1

Figure 1.21: Missing Values Represented Based on Variable Type

Length

A variable's length (the number of bytes used to store it) is related to its type.

- Character variables can be up to 32,767 bytes long. In the example below, Name has a length of 20 characters and uses 20 bytes of storage.
- All numeric variables have a default length of 8 bytes. Numeric values (no matter how many digits they contain) are stored as floating-point numbers in 8 bytes of storage.

Table 1.8: Length Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

You've seen that each SAS variable has a name, type, and length. In addition, you can optionally define format, informat, and label attributes for variables. Let's look briefly at these optional attributes—you'll learn more about them in later chapters as you need to use them.

Format

Formats are variable attributes that affect the way data values are written. SAS software offers a variety of character, numeric, and date and time formats. You can also create and store your own formats. To write values out using a particular form, you select the appropriate format.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

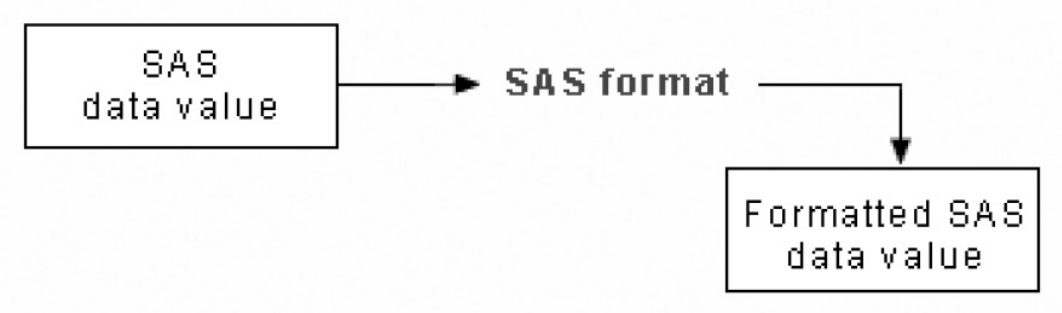


Figure 1.22: Formats

For example, to display the value **1234** as $1,234.00 in a report, you can use the DOLLAR8.2 format, as shown for Total below.

Table 1.9: Format Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Usually you have to specify the maximum width (w) of the value to be written. Depending on the particular format, you may also need to specify the number of decimal places (d) to be written. For example, to display the value **5678** as 5,678.00 in a report, you can use the COMMA8.2 format, which specifies a width of 8 including 2 decimal places.

Additional Note You can permanently assign a format to a variable in a SAS data set, or you can temporarily specify a format in a PROC step to determine the way the data values appear in output.

Informat

Whereas formats write values out using some particular form, informats read data values in certain forms into standard SAS values. Informats determine how data values are read into a SAS data set. You *must* use informats to read numeric values that contain letters or other special characters.

Figure 1.23: Informats

For example, the numeric value **$12,345.00** contains two special characters, a dollar sign ($) and a comma (,). You can use an informat to read the value while removing the dollar sign and comma, and then store the resulting value as a standard numeric value. For Total below, the COMMA10. informat is specified.

Table 1.10: Informat Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Label

A variable can have a label, which consists of descriptive text up to 256 characters long. By default, many reports identify variables by their names. You may want to display more descriptive information about the variable by assigning a label to the variable.

For example, you can label Policy as Policy Number, Total as Total Balance, and Name as Patient Name to display these labels in reports.

Table 1.11: Label Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

You may even want to use labels to shorten long variable names in your reports!

Using the Programming Workspace

Using the Main SAS Windows

When you start SAS, by default several primary windows are available to you. These include the Explorer, Log, Output, Results, and code editing window(s). The window you use to edit your SAS programs may vary, depending on your operating system and needs.

You use these SAS windows to explore and manage your files, to enter and submit SAS programs, to view messages, and to view and manage your output.

We'll tour each of these windows shortly.

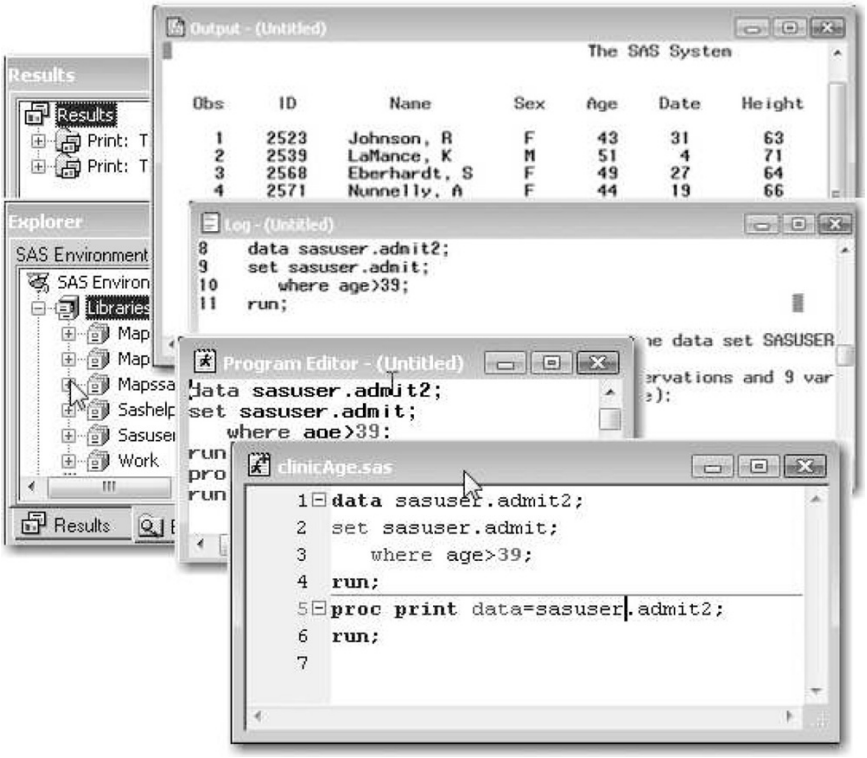


Figure 1.24: Using the Main SAS Windows

Your operating environment, and any options that you use when you start SAS, determine

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

- which of the main SAS windows are displayed by default
- their general appearance
- their position.

Features of SAS Windows

SAS windows have many features that help you get your work done. For example, you can

- maximize, minimize, and restore windows
- use pull-down menus, pop-up menus, and toolbars
- get more help.

Additional Note This chapter and later chapters show SAS®9 windows in the Windows operating environment.

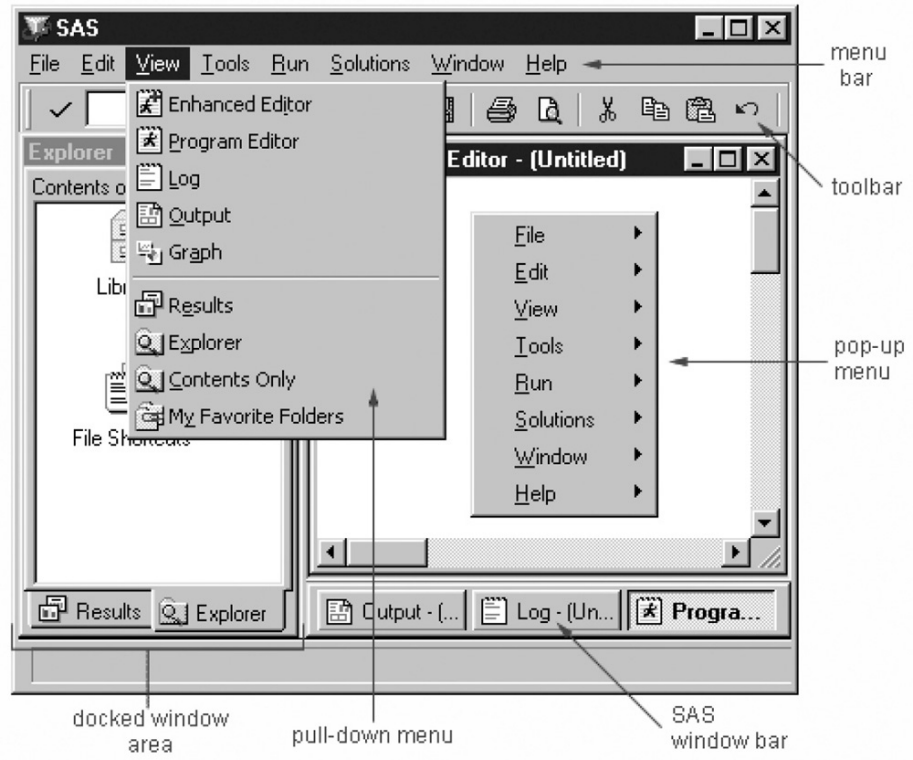


Figure 1.25: Features of SAS Windows

Minimizing and Restoring Windows

In the Windows environment, you can click the **Minimize** button to send a window that you aren't using to the SAS window bar. To restore the window to its former position, click the corresponding button on the SAS window bar.

In other operating environments, minimizing the window shrinks it to an icon.

Docking and Undocking Windows

In the Windows and OS/2 environments, the Explorer and Results windows are docked by default, so they can be resized but not minimized. If you prefer, you can select **Window** ⇨ **Docked** to undock the active window, or you can turn docking off completely in the Preferences dialog box.

Issuing Commands

In SAS, you can issue commands by

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

- making selections from a menu bar
- by typing commands in a command box (or Toolbox) or on a command line.

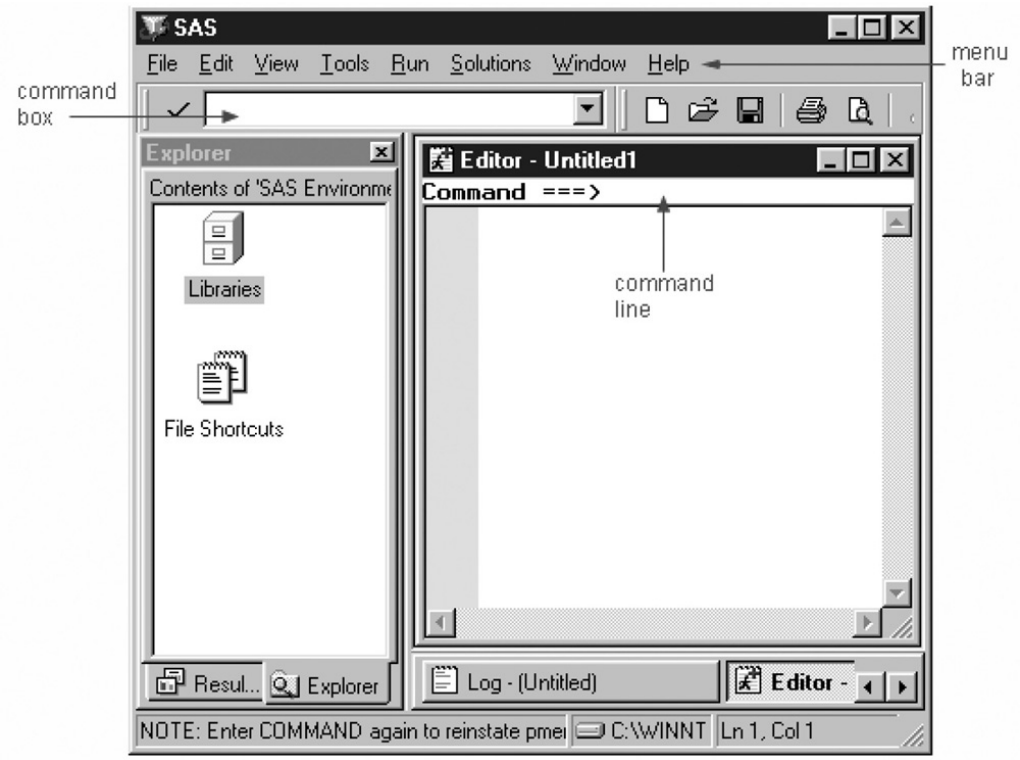


Figure 1.26: Using the SAS Command Box and Menu Bar to Issue Commands

In most operating environments, SAS displays a menu bar by default. In the Windows environment, the menu bar selections correspond to the active window. To display a menu bar if it is not displayed, you can type **pmenus** in the command box (or ToolBox) or on the command line.

In all operating environments, SAS displays a command box (or ToolBox) or command line by default. You can display a command line in a particular window by activating that window and then using the Tools menu as indicated below:

- In the Windows environment, select **Tools** ⇨ **Options** ⇨ **Preferences**, and then select the **View** tab and select the **Command line** checkbox.
- In the UNIX and z/OS environments, select **Tools** ⇨ **Options** ⇨ **Turn Command Line On**.
- In the z/OS environment, you can display both a command line and a menu bar simultaneously in a window by selecting **Tools** ⇨ **Options** ⇨ **Command…**.

In the Windows and UNIX operating environments, you can also display a command line in a window by activating the window, typing **command** in the command box (or ToolBox), and pressing Enter.

Additional Note See the online help for a complete list of command-line commands.

Using Pop-Up Menus

Pop-up menus are context sensitive; they list actions that pertain only to a particular window. Generally, you display pop-up menus by clicking the right mouse button. If you like, you can specify a function key to open pop-up menus. Simply select **Tools** ⇨ **Options** ⇨ **Keys** and type **wpopup** as a function key setting.

To open a pop-up menu in the z/OS operating environment, type **?** in the selection field beside the item.

Getting Help

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Help is available for all windows in SAS. From the **Help** menu, you can access comprehensive online help and documentation for SAS, or you can access task-oriented help for the active window. The Help menu is discussed in more detail later in this chapter.

Customizing Your SAS Environment

You can customize many features of the SAS workspace such as toolbars, pop-up menus, icons, and so on. Select the **Tools** menu to explore some of the customization options that are available.

You'll learn how to use features of SAS windows throughout this chapter. Now let's look at each of the main SAS windows individually.

The Explorer Window

In the Explorer window, you can view and manage your SAS files, which are stored in SAS data libraries. The library name is a logical name for the physical location of the files (such as a directory). You can think of the library name as a temporary nickname or shortcut.

You use the Explorer window to

- create new libraries and SAS files
- open any SAS file
- perform most file management tasks such as moving, copying, and deleting files
- create shortcuts to files that were not created with SAS.

Notice that the Explorer window displays a tree view of its contents.

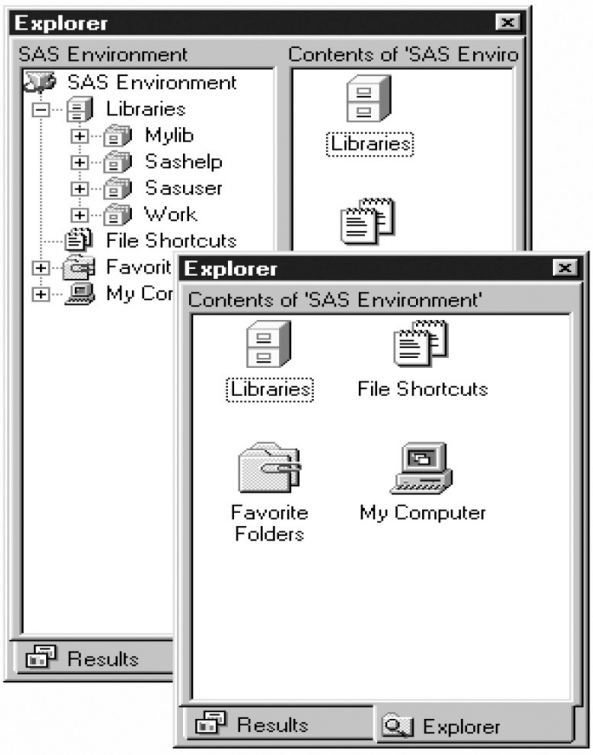


Figure 1.27: Tree View of the SAS Explorer Window

You can display the Explorer window by selecting **View** ⇨ **Explorer**. In the Windows and z/OS operating environments, if the Explorer window is docked, you can click the **Explorer** tab to display the window.

Navigating the Explorer Window

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

You can find your way around the Explorer window by double-clicking folders to open them and see their contents. You can also use pop-up menus to perform actions on a file (such as viewing its properties, or copying it). Pop-up menus contain different options for different file types.

To open a pop-up menu in the z/OS operating environment, type **?** in the selection field beside the item in the Explorer window. To simulate a double-click, type **s** in the selection field beside the item.

Code Editing Windows

You can use the following editors to write and edit SAS programs:

- the Enhanced Editor window
- the Program Editor window
- the host editor of your choice.

This training focuses on the two SAS code editing windows: the Enhanced Editor and the Program Editor windows. The Enhanced Editor is available only in the Windows operating environment.

The features of both editors are described below. In the remaining chapter and in future chapters, the general term code editing window will be used to refer to your preferred SAS code editing window.

Enhanced Editor Window

In the Windows operating environment, an Enhanced Editor window opens by default. You can use the Enhanced Editor window to enter, edit, and submit SAS programs. The initial window title is Editor - Untitled *n* until you open a file or save the contents of the editor to a file. Then the window title changes to reflect that filename. When the contents of the editor are modified, an asterisk is added to the title.

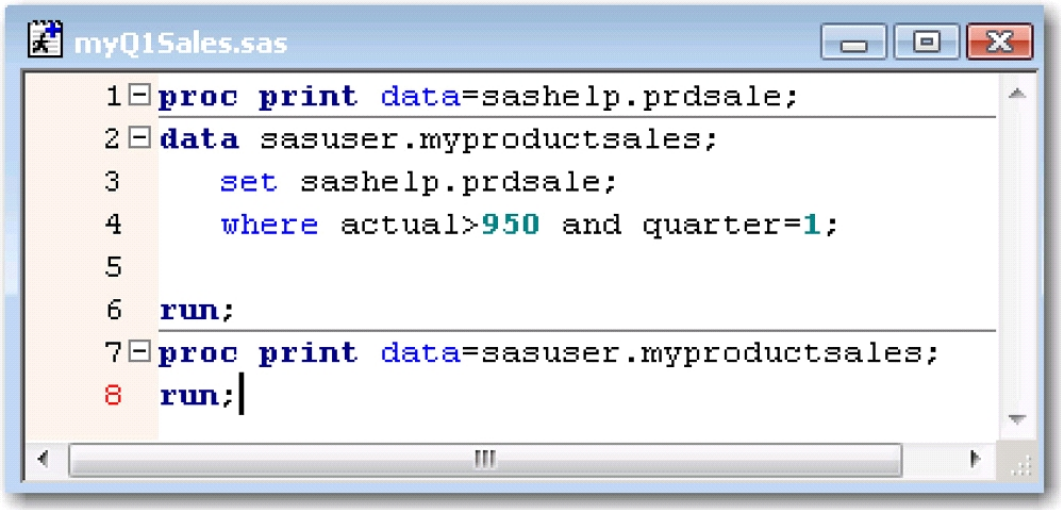


Figure 1.28: Enhanced Editor Window

You can redisplay or open additional Enhanced Editor windows by selecting **View** ⇨ **Enhanced Editor**.

Enhanced Editor Features

In the Enhanced Editor, you can perform standard editing tasks such as

- opening SAS programs in various ways, including drag and drop
- entering, editing, and submitting SAS programs
- using the command line or menus
- saving SAS programs

Reprinted for OET7P/856094, Accenture
SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

- clearing contents.

In addition, the Enhanced Editor provides useful editing features, including

- color coding and syntax checking of the SAS programming language
- expandable and collapsible sections
- recordable macros
- support for keyboard shortcuts (Alt or Shift plus keystroke)
- multi-level undo and redo.

For more information about the Enhanced Editor, open or activate the Enhanced Editor window, and then select **Help** ⇨ **Using This Window**.

Clearing the Editor

In the Enhanced Editor, the code does not disappear when you submit it.

To clear any of these windows, you can activate the window and select **Edit** ⇨ **Clear All**.

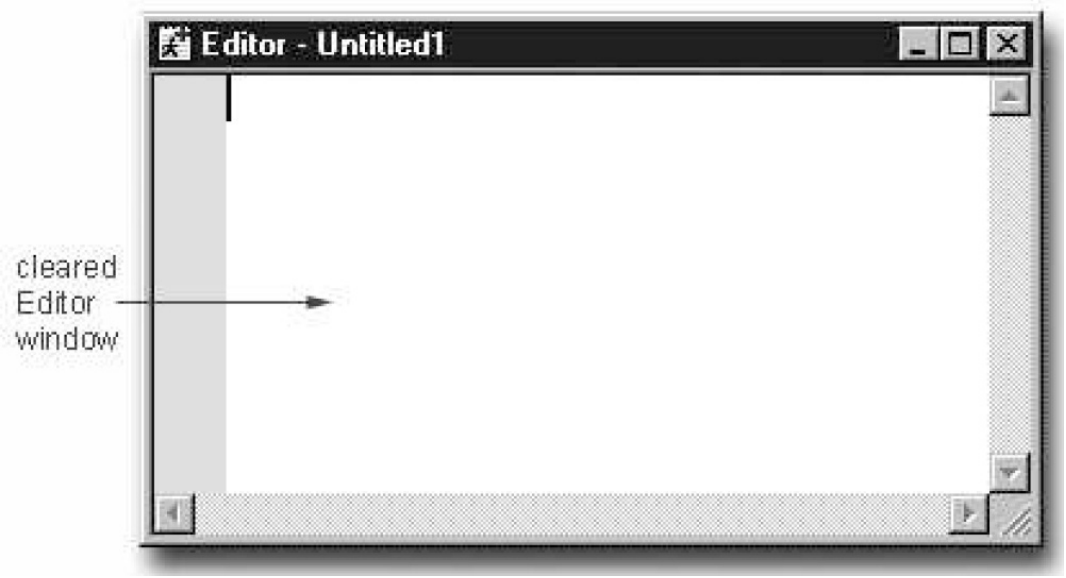


Figure 1.29: Cleared Editor Window

The Program Editor Window

As in the Enhanced Editor window, in the Program Editor window you enter, edit, and submit SAS programs. You can also open existing SAS programs. You can display the Program Editor window by selecting **View** ⇨ **Program Editor**.

Reprinted for OET7P/856094, Accenture
SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

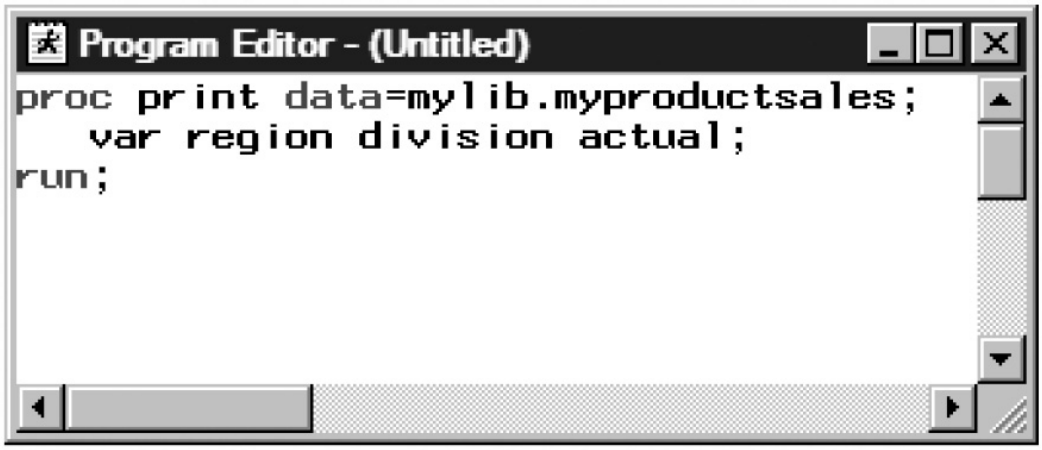


Figure 1.30: Program Editor Window

Features

As in the Enhanced Editor window, in the Program Editor window you can perform standard editing tasks such as

- opening SAS programs in various ways, including drag and drop
- entering, editing, and submitting SAS programs
- using the command line or menus
- saving SAS programs
- clearing contents
- recalling submitted statements.

However, the Program Editor does not provide some features of the enhanced Editor, such as syntax checking (this feature is available in SAS 9.2), expandable and collapsible sections, and recordable macros.

For more information about these features, open or activate the Program Editor window, and then select **Help** ⇨ **Using This Window**.

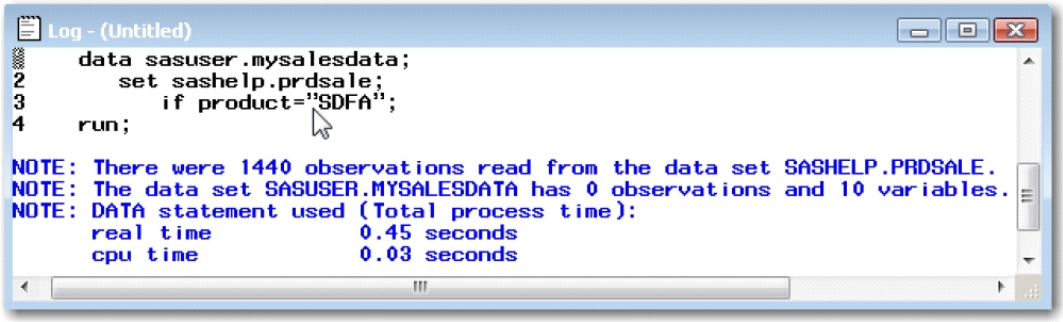
Clearing the Editor

At any time you can clear program code from the Program Editor window by activating the window and selecting **Edit** ⇨ **Clear All**.

When you submit SAS programs in the Program Editor window, the code in the window is automatically cleared.

The Log Window

The Log window displays messages about your SAS session and about any SAS programs that you submit. You can display the Log window by selecting **View** ⇨ **Log**.



Reprinted for OET7P/856094, Accenture
SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Figure 1.31: Log Window

The Output Window

In the Output window, you browse LISTING output from SAS programs that you submit. (You can use a browser to view HTML output.)

By default, the Output window is positioned behind the code editing and Log windows. When you create output, the Output window automatically moves to the front of your display.

You can display the Output window at any time by selecting **View** ⇨ **Output**.

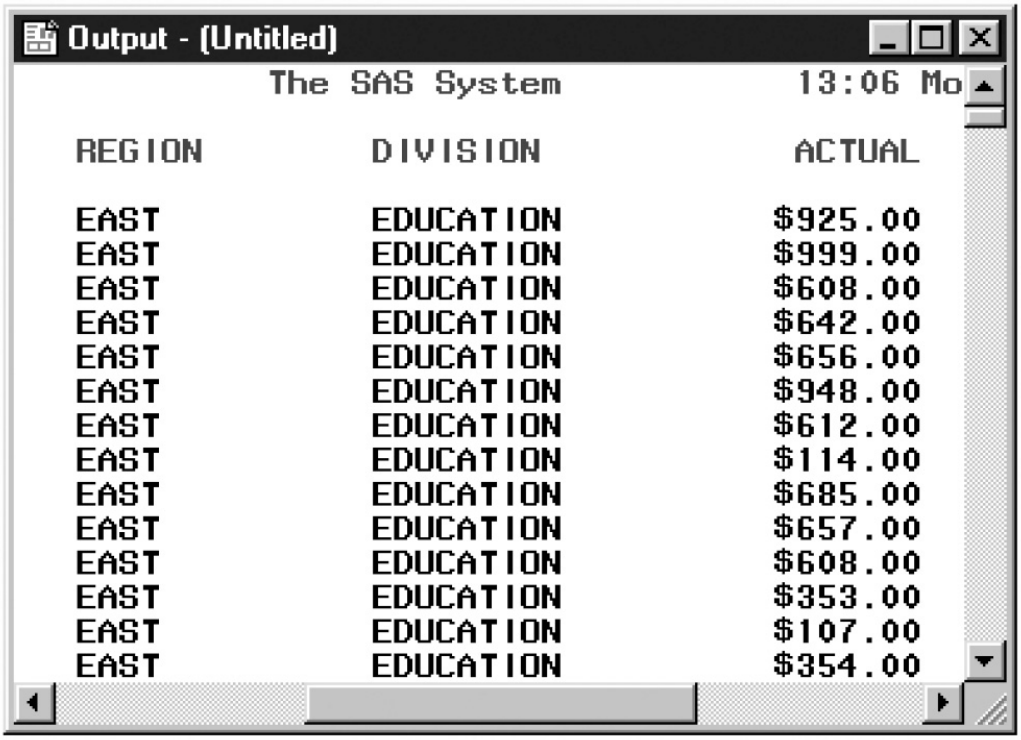


Figure 1.32: Output Window

Not all SAS programs create output in the Output window. Some open interactive windows. Others produce only messages in the Log window.

Caution In mainframe operating environments, when you create multiple pages of output, a message in the Output window border indicates that the procedure is suspended. In the example below, PROC PRINT output is suspended.

Figure 1.33: Mainframe Window Showing Suspended Procedure

To remove the message and view the remaining output, simply scroll to the bottom of the output.

The Results Window

The Results window helps you navigate and manage output from SAS programs that you submit. You can view, save, and print individual items of output. The Results window uses a tree structure to list various types of output that might be available after you run SAS.

On most operating systems, the Results window is positioned behind the Explorer window and is empty until you submit a

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

SAS program that creates output. Then the Results window moves to the front of your display. The Results window displays separate icons for LISTING output and HTML output. In the example below, the first Print folder contains both types of output.

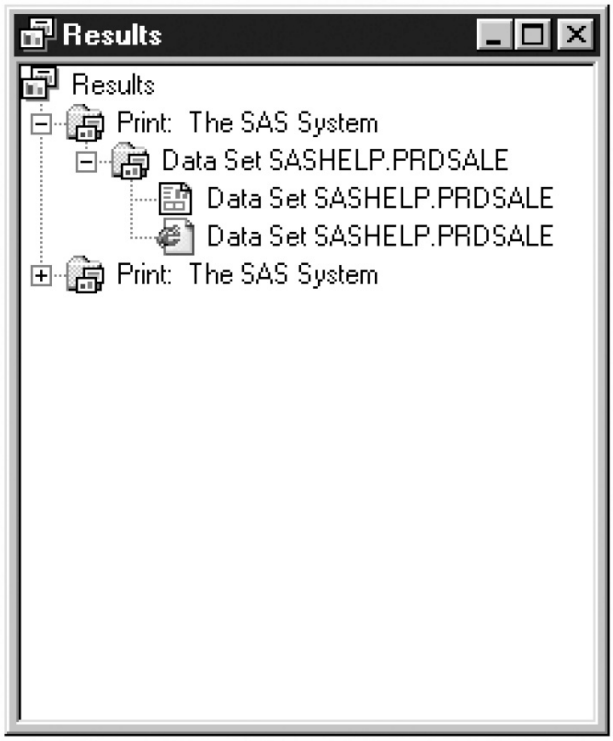


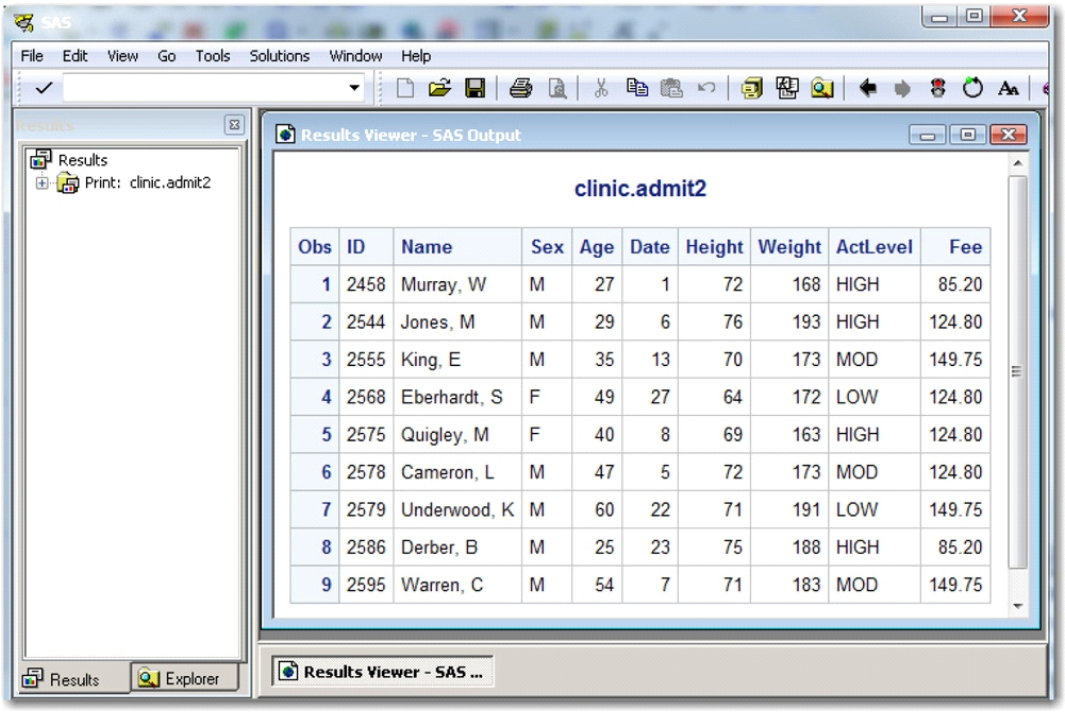
Figure 1.34: Results Window

Viewing Output in the Results Window

You can display the Results window at any time by selecting **View** ⇨ **Results**. HTML is the default output type in the SAS windowing environment for UNIX and Windows. In these environments, when you submit a SAS program, the HTML output is automatically displayed in the Results Viewer and the file is listed in the Results window. In all other environments, LISTING is the default output.

The left pane of the following display shows the Results window, and the right pane shows the Results Viewer where the default HTML output is displayed. The Results window lists the files that were created when the SAS program executed.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited



Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2544	Jones, M	M	29	6	76	193	HIGH	124.80
3	2555	King, E	M	35	13	70	173	MOD	149.75
4	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80
5	2575	Quigley, M	F	40	8	69	163	HIGH	124.80
6	2578	Cameron, L	M	47	5	72	173	MOD	124.80
7	2579	Underwood, K	M	60	22	71	191	LOW	149.75
8	2586	Derber, B	M	25	23	75	188	HIGH	85.20
9	2595	Warren, C	M	54	7	71	183	MOD	149.75

Figure 1.35: Results Window and Results Viewer

Creating SAS Libraries

Earlier in this chapter, you saw that SAS files are stored in libraries. By default, SAS defines several libraries for you (including Sashelp, Sasuser, and Work). You can also define additional libraries.

Sashelp

a permanent library that contains sample data and other files that control how SAS works at your site. This is a read-only library.

Sasuser

a permanent library that contains SAS files in the Profile catalog that store your personal settings. This is also a convenient place to store your own files.

Work

a temporary library for files that do not need to be saved from session to session.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

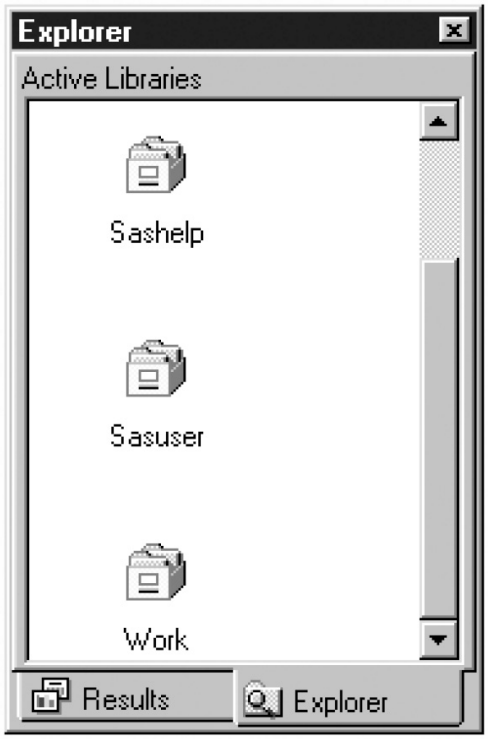


Figure 1.36: Active SAS Libraries

You can also *define* additional libraries. When you define a library, you indicate the location of your SAS files to SAS. Once you define a library, you can manage SAS files within it.

When you *delete* a SAS library, the pointer to the library is deleted, and SAS no longer has access to the library. However, the contents of the library still exist in your operating environment.

Defining Libraries

To define a library, you assign a library name to it and specify a path, such as a directory path. (In some operating environments you must create the directory or other storage location before defining the library.)

You can also specify an engine, which is a set of internal instructions that SAS uses for writing to and reading from files in a library.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

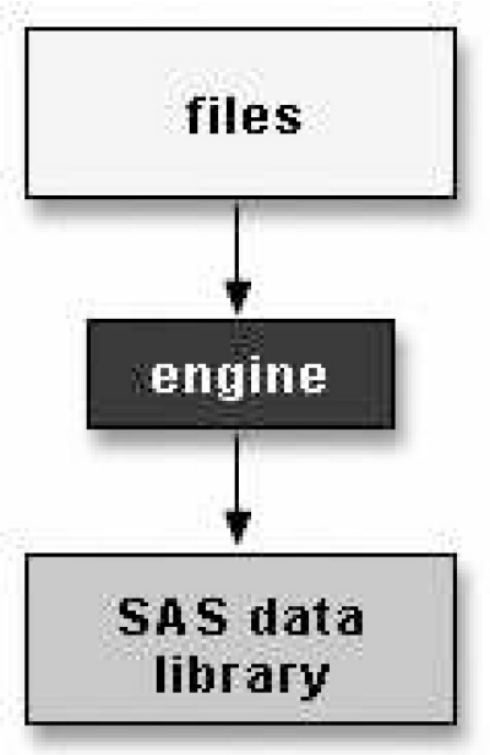


Figure 1.37: Defining LIbraries

Additional Note In this chapter, you learn about SAS libraries. You can define SAS libraries using programming statements. "Specifying Engines" on page 47 shows you how to write LIBNAME statements to define SAS libraries.

Additional Note Depending on your operating environment and the SAS/ACCESS products that you license, you can create libraries with various engines. Each engine enables you to read a different file format, including file formats from other software vendors.

Creating and Using File Shortcuts

You've seen that the Explorer window gives you access to your SAS files. You can also create a file shortcut to an external file.

An external file is a file that is created and maintained in your host operating environment. External files contain data or text, such as

- SAS programming statements
- records of raw data
- procedure output.

SAS can use external files, but they are not managed by SAS.

A file shortcut (or *fileref)* is an optional name that is used to identify an external file to SAS. File shortcuts are stored in the File Shortcuts folder in the Explorer window. You can use a file shortcut to open, browse, and submit a file.

When you delete a file shortcut, the pointer to the file is deleted, and SAS no longer has access to the file. However, the file still exists in your operating environment.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

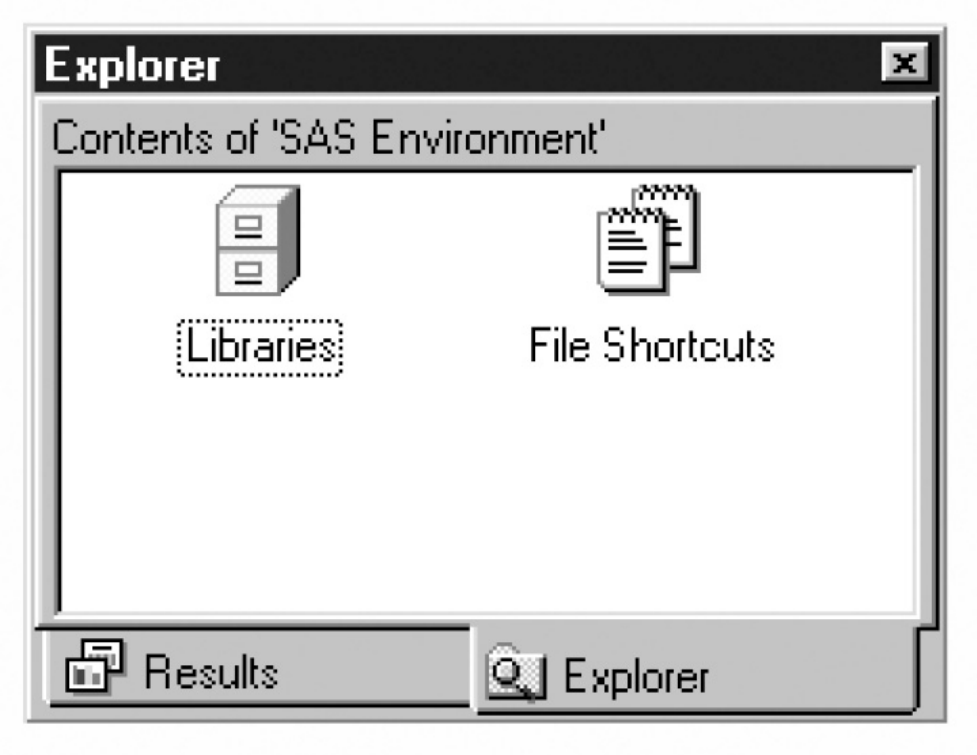


Figure 1.38: Creating and Using File Shortcuts

Additional Note If you have used SAS before, a file shortcut is the same as a *file reference* or *fileref*.

Using SAS Solutions and Tools

Along with windows for working with your SAS files and SAS programs, SAS provides a set of ready-to-use solutions, applications, and tools. You can access many of these tools by using the Solutions and Tools menus.

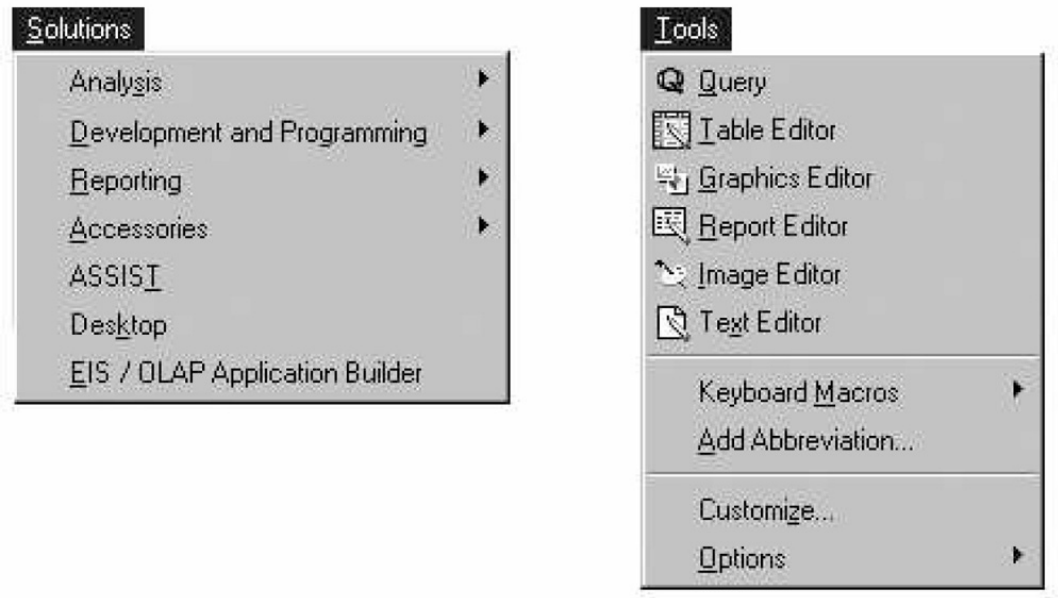


Figure 1.39: Using SAS Solutions and Tools

For example, you can use the table editor in the Tools menu to enter, browse, or edit data in a SAS data set.

Getting Help

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

You've learned to use SAS windows to perform common SAS tasks. As you begin working in SAS, be sure to take advantage of the different types of online help that are available from the Help menu.

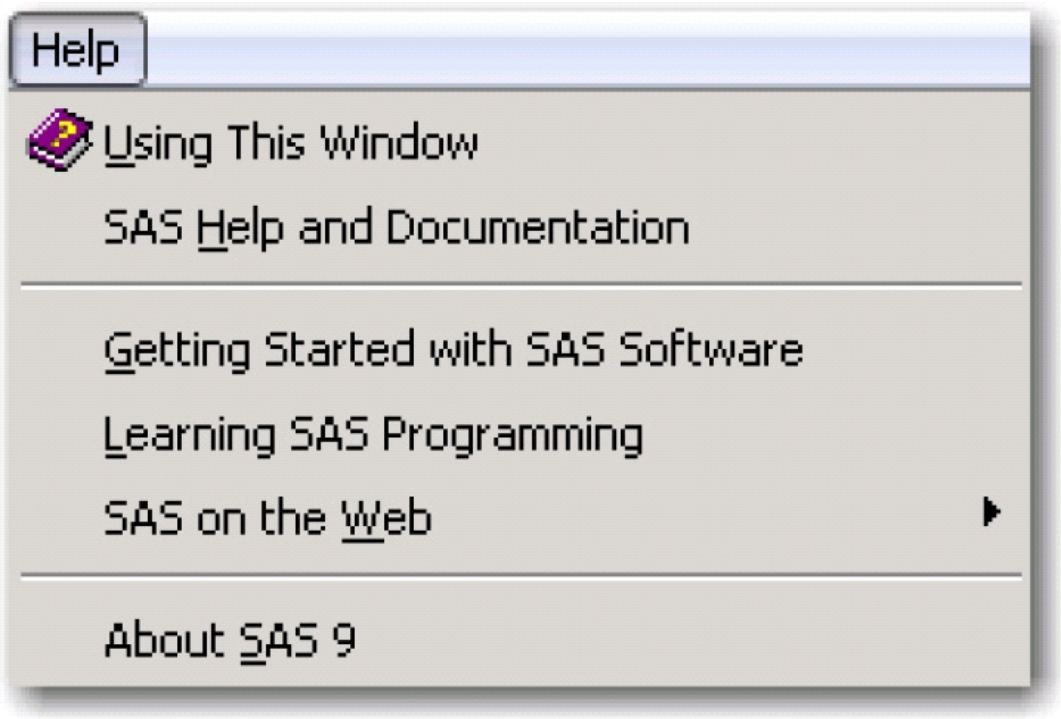


Figure 1.40: SAS Help

- **Using This Window** is task-oriented help for the active window.
- **SAS Help and Documentation** is a complete guide to syntax, examples, procedures, concepts, and what's new.
- **Getting Started tutorials** are listed under Help for products where they are available.
- Selecting **Learning SAS programming** enables you to create data that is used in online training courses, and displays SAS OnlineTutor if you have a site license.
- If you have Internet access, **SAS on the Web** provides links to information including Technical Support and Frequently Asked Questions.

Additional Note To access SAS online help, documentation, and other resources from your SAS software, select **Help** ⇨ **SAS Documentation** from the SAS toolbar. You can also access SAS documentation in the SAS Knowledge Base at support.sas.com.

Chapter Summary

Text Summary

Components of SAS Programs

SAS programs consist of two types of steps: DATA steps and PROC (procedure) steps. These two steps, alone or combined, form most SAS programs. A SAS program can consist of a DATA step, a PROC step, or any combination of DATA and PROC steps. DATA steps typically create or modify SAS data sets, but they can also be used to produce custom-designed reports. PROC steps are pre-written routines that enable you to analyze and process the data in a SAS data set and to present the data in the form of a report. They sometimes create new SAS data sets that typically contain the results of the procedure.

Characteristics of SAS Programs

SAS programs consist of SAS statements. A SAS statement usually begins with a SAS keyword and always ends with a semicolon. A DATA step begins with the keyword DATA. A PROC step begins with the keyword PROC. SAS statements

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

are free-format, so they can begin and end anywhere on a line. One statement can continue over several lines, and several statements can be on a line. Blanks or special characters separate "words" in a SAS statement.

Processing SAS Programs

When you submit a SAS program, SAS reads SAS statements and checks them for errors. When it encounters a subsequent DATA, PROC, or RUN statement, SAS executes the previous step in the program.

Each time a step is executed, SAS generates a log of the processing activities and the results of the processing. The SAS log collects messages about the processing of SAS programs and about any errors that occur.

The results of processing can vary. Some SAS programs open an interactive window or invoke procedures that create output in the form of a report. Other SAS programs perform tasks such as sorting and managing data, which have no visible results other than messages in the log.

SAS Libraries

Every SAS file is stored in a SAS library, which is a collection of SAS files such as SAS data sets and catalogs. In the Windows and UNIX environments, a SAS library is typically a group of SAS files in the same folder or directory.

Depending on the libref you use, you can store SAS files in a temporary SAS library or in permanent SAS libraries.

- Temporary SAS files that are created during the session are held in a special work space that is assigned the default libref Work. If you don't specify a libref when you create a file (or if you specify Work), the file is stored in the temporary SAS library. When you end the session, the temporary library is deleted.
- To store files permanently in a SAS library, you assign it a libref other than the default Work. For example, by assigning the libref sasuser to a SAS library, you specify that files within the library are to be stored until you delete them.

Referencing SAS Files

To reference a SAS file, you use a two-level name, *libref.filename*. In the two-level name, *libref* is the name for the SAS library that contains the file, and *filename* is the name of the file itself. A period separates the libref and filename.

To reference temporary SAS files, you specify the default libref Work, a period, and the filename. Alternatively, you can simply use a one-level name (the filename only) to reference a file in a temporary SAS library. Referencing a SAS file in any library *except Work* indicates that the SAS file is stored permanently.

SAS data set names can be 1 to 32 characters long, must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_), and can continue with any combination of numerals, letters, or underscores.

SAS Data Sets

For many of the data processing tasks that you perform with SAS, you access data in the form of a SAS data set and use SAS programs to analyze, manage, or present the data. Conceptually, a SAS data set is a file that consists of two parts: a descriptor portion and a data portion. Some SAS data sets also contain one or more indexes, which enable SAS to locate records in the data set more efficiently.

The descriptor portion of a SAS data set contains property information about the data set.

The data portion of a SAS data set is a collection of data values that are arranged in a rectangular table. Observations in the data set correspond to rows or data lines. Variables in the data set correspond to columns. If a data value is unknown for a particular observation, a missing value is recorded in the SAS data set.

Variable Attributes

In addition to general information about the data set, the descriptor portion contains property information for each variable in the data set. The property information includes the variable's name, type, and length. A variable's type determines how missing values for a variable are displayed by SAS. For character variables, a *blank* represents a missing value. For numeric variables, a *period* represents a missing value. You can also specify format, informat, and label properties for variables.

Using the Main SAS Windows

You use the following windows to explore and manage your files, to enter and submit SAS programs, to view messages,

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

and to view and manage your output.

Table 1.12: Windows and How They Are Used

Use this window …	To …
Explorer	view your SAS files create new libraries and SAS files perform most file management tasks such as moving, copying, and deleting files create shortcuts to files that were not created with SAS
Enhanced Editor (code editing window)	enter, edit, and submit SAS program **Additional Note** The Enhanced Editor window is available only in the Windows operating environment.
Program Editor (code editing window)	enter, edit, and submit SAS programs
Log	view messages about your SAS session and about any SAS programs that you submit
Output	browse output from SAS programs
Results	navigate and manage output from SAS programs view, save, and print individual items of output

Points to Remember

- Before referencing SAS files, you must assign a name (libref, or library reference) to the library in which the files are stored (or specify that SAS is to assign the name automatically).
- You can store SAS files either temporarily or permanently.
- Variable names follow the same rules as SAS data set names. However, your site may choose to restrict variable names to those valid in Version 6 SAS, to uppercase variable names automatically, or to remove all restrictions on variable names.

Chapter Quiz

Select the best answer for each question. After completing the quiz, you can check your answers using the answer key in the appendix.

1. How many observations and variables does the data set below contain? ?

Name	Sex	Age
Picker	M	32
Fletcher		28
Romano	F	•
Choi	M	42

a. 3 observations, 4 variables

b. 3 observations, 3 variables

c. 4 observations, 3 variables

d. can't tell because some values are missing

2. How many program steps are executed when the program below is processed? ?

```
data user.tables;
   infile jobs;
   input date yyddmm8. name $ job $;
run;
proc sort data=user.tables;
   by name;
run;
proc print data=user.tables;
run;
```

a. three

b. four

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

c. five

d. six

3. What type of variable is the variable AcctNum in the data set below? ?

AcctNum	Gender
3456_1	M
2451_2	
Romano	F
Choi	M

a. numeric

b. character

c. can be either character or numeric

d. can't tell from the data shown

4. What type of variable is the variable Wear in the data set below? ?

Brand	Wear
Acme	43
Ajax	34
Atlas	•

a. numeric

b. character

c. can be either character or numeric

d. can't tell from the data shown

5. Which of the following variable names is valid? ?

a. 4BirthDate

b. $Cost

c. _Items_

d. Tax-Rate

6. Which of the following files is a permanent SAS file? ?

a. Sashelp.PrdSale

b. Sasuser.MySales

c. Profits.Quarter1

d. all of the above

7. In a DATA step, how can you reference a temporary SAS data set named Forecast? ?

a. Forecast

b. Work.Forecast

c. Sales.Forecast (after assigning the libref Sales)

d. only a and b above

8. What is the default length for the numeric variable Balance? ?

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Name	Balance
Adams	105.73
Geller	107.89
Martinez	97.45
Noble	182.50

a. 5

b. 6

c. 7

d. 8

9. How many statements does the following SAS program contain? ?

```
proc print data=new.prodsale
                label double;
   var state day price1 price2; where state='NC';
   label state='Name of State'; run;
```

a. three

b. four

c. five

d. six

10. What is a SAS library? ?

a. collection of SAS files, such as SAS data sets and catalogs

b. in some operating environments, a physical collection of SAS files

c. a group of SAS files in the same folder or directory

d. all of the above

Answers

1. Correct answer: c

Rows in the data set are called observations, and columns are called variables. Missing values don't affect the structure of the data set.

2. Correct answer: a

When it encounters a DATA, PROC, or RUN statement, SAS stops reading statements and executes the previous step in the program. The program above contains one DATA step and two PROC steps, for a total of three program steps.

3. Correct answer: b

It must be a character variable, because the values contain letters and underscores, which are not valid characters for numeric values.

4. Correct answer: a

It must be a numeric variable, because the missing value is indicated by a period rather than by a blank.

5. Correct answer: c

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited

Variable names follow the same rules as SAS data set names. They can be 1 to 32 characters long, must begin with a letter (A-Z, either uppercase or lowercase) or an underscore, and can continue with any combination of numerals, letters, or underscores.

6. Correct answer: d

To store a file permanently in a SAS data library, you assign it a libref other than the default Work. For example, by assigning the libref Profits to a SAS data library, you specify that files within the library are to be stored until you delete them. Therefore, SAS files in the Sashelp and Sasuser and Profits libraries are permanent files.

7. Correct answer: d

To reference a temporary SAS file in a DATA step or PROC step, you can specify the one-level name of the file (for example, Forecast) or the two-level name using the libref Work (for example, Work.Forecast).

8. Correct answer: d

The numeric variable Balance has a default length of 8. Numeric values (no matter how many digits they contain) are stored in 8 bytes of storage unless you specify a different length.

9. Correct answer: c

The five statements are: 1) the PROC PRINT statement (two lines long); 2) the VAR statement; 3) the WHERE statement (on the same line as the VAR statement); 4) the LABEL statement; and 5) the RUN statement (on the same line as the LABEL statement).

10. Correct answer: d

Every SAS file is stored in a SAS library, which is a collection of SAS files, such as SAS data sets and catalogs. In some operating environments, a SAS library is a physical collection of files. In others, the files are only logically related. In the Windows and UNIX environments, a SAS library is typically a group of SAS files in the same folder or directory.

Reprinted for OET7P/856094, Accenture SAS Institute, SAS Institute Inc (c) 2011, Copying Prohibited