1

Essential Concepts

What Is the SAS System?

The SAS System is an integrated system of software products that enables you to perform

- data entry, retrieval, and management
- report writing and graphics
- statistical and mathematical analysis
- business planning, forecasting, and decision support
- operations research and project management
- quality improvement
- applications development.

In addition, you can integrate with SAS many SAS business solutions that enable you to perform large scale business functions, such as data warehousing and data

mining, human resources management and decision support, financial management and decision support, and others.

Overview of Base SAS Software

The core of the SAS System is base SAS software, which consists of

SAS language	a programming language that you use to manage your data.
SAS procedures	software tools for data analysis and reporting.
macro facility	a tool for extending and customizing SAS software programs and for reducing text in your programs.
DATA step debugger	a programming tool that helps you find logic problems in DATA step programs.
Output Delivery System (ODS)	a system that delivers output in a variety of easy-to-access formats, such as SAS data sets, listing files, or Hypertext Markup Language (HTML).
SAS windowing environment	an interactive, graphical user interface that enables you to easily run and test your SAS programs.

This document, when used with *SAS Language Reference: Dictionary*, covers only the SAS language. For a complete guide to base SAS software, also see these documents: *SAS Procedures Guide*, *SAS Macro Language Dictionary*, and *Getting Started with the SAS System*. The SAS windowing environment is described in the online Help.

Components of the SAS Language

SAS Files

When you work with SAS, you use files that are created and maintained by SAS, as well as files that are created and maintained by your operating environment, and that are not related to SAS. Files with formats or structures known to SAS are referred to as *SAS files*. All SAS files reside in a *SAS data library*.

The most commonly used SAS file is a *SAS data set*. A SAS data set is structured in a format that SAS can process. Another common type of SAS file is a *SAS catalog*. Many different kinds of information that are used in a SAS job are stored in SAS catalogs, such as instructions for reading and printing data values, or function key settings that you use in the SAS windowing environment. A *SAS stored program* is a type of SAS file that contains compiled code that you create and save for repeated use.

Operating Environment Information: In some operating environments, a SAS data library is a physical relationship among files; in others, it is a logical relationship. Refer to the SAS documentation for your operating environment for details about the characteristics of SAS data libraries in your operating environment. △

SAS Data Sets
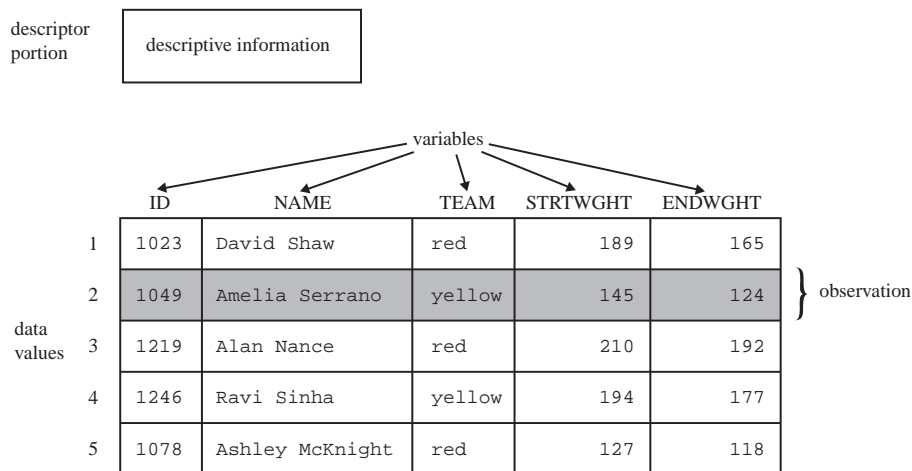
There are two kinds of SAS data sets:

☐ SAS data file

□ SAS data view.

A *SAS data file* both describes and physically stores your data values. A *SAS data view*, on the other hand, does not actually store values. Instead, it is a query that creates a logical SAS data set that you can use as if it were a single SAS data set. It enables you to look at data stored in one or more SAS data sets or in other vendors' software files. SAS data views enable you to create logical SAS data sets without using the storage space required by SAS data files.

A SAS data set consists of the following:

□ descriptor information

□ data values.

The descriptor information describes the contents of the SAS data set to SAS. The *data values* are data that has been collected or calculated. They are organized into rows, called observations, and columns, called variables. An *observation* is a collection of data values that usually relate to a single object. A *variable* is the set of data values that describe a given characteristic. The following figure represents a SAS data set:

		ID	NAME	TEAM	STRTWGHT	ENDWGHT
	1	1023	David Shaw	red	189	165
	2	1049	Amelia Serrano	yellow	145	124
	3	1219	Alan Nance	red	210	192
	4	1246	Ravi Sinha	yellow	194	177
	5	1078	Ashley McKnight	red	127	118

descriptor portion — descriptive information

variables

data values

} observation

Usually, an observation is the data that is associated with an entity such as an inventory item, a regional sales office, a client, or a patient in a medical clinic. Variables are characteristics of these entities, such as sale price, number in stock, and originating vendor. When data values are incomplete, SAS uses a *missing value* to represent a missing variable within an observation.

External Files

Data files that you use to read and write data, but which are in a structure unknown to SAS, are called *external files*. External files can be used for storing

□ raw data that you want to read into a SAS data file

□ SAS program statements

□ procedure output.

Operating Environment Information: Refer to the SAS documentation for your operating environment for details about the characteristics of external files in your operating environment. △

Database Management System Files

SAS software is able to read and write data to and from other vendors' software, such as many common database management system (DBMS) files. In addition to base SAS software, you must license the SAS/ACCESS software for your DBMS and operating environment.

SAS Language Elements

The SAS language consists of statements, expressions, options, formats, and functions similar to those of many other programming languages. In SAS, you use these elements within one of two groups of SAS statements:

- ☐ DATA steps
- ☐ PROC steps.

A *DATA step* consists of a group of statements in the SAS language that reads raw data or existing SAS data sets to create a SAS data set. Once your data is accessible as a SAS data set, you can analyze the data and write reports by using a set of tools known as SAS procedures.

A group of procedure statements is called a *PROC step*. SAS procedures analyze data in SAS data sets to produce statistics, tables, reports, charts, and plots, to create SQL queries, and to perform other analyses and operations on your data. They also provide ways to manage and print SAS files.

You can also use global SAS statements and options outside of a DATA step or PROC step.

SAS Macro Facility

Base SAS software includes the SAS Macro Facility, a powerful programming tool for extending and customizing your SAS programs, and for reducing the amount of code that you must enter to do common tasks. Macros are SAS files that contain compiled macro program statements and stored text. You can use macros to automatically generate SAS statements and commands, write messages to the SAS log, accept input, or create and change the values of macro variables. For complete documentation, see *SAS Macro Language: Reference*.

Running SAS

Starting a SAS Session

You start a SAS session with the SAS command, which follows the rules for other commands in your operating environment. In some operating environments, you include the SAS command in a file of system commands or control statements; in others, you enter the SAS command at a system prompt or select SAS from a menu.

Different Types of SAS Sessions

You can run SAS in any of several different ways that might be available for your operating environment:

□ SAS windowing environment

□ interactive line mode

□ noninteractive mode

□ batch (or background) mode.

In addition, SAS/ASSIST software provides a menu-driven system for creating and running your SAS programs. For more information about SAS/ASSIST, see *Getting Started with the SAS System Using SAS/ASSIST Software*.

SAS Windowing Environment

In the *SAS windowing environment*, you can edit and execute programming statements, display the SAS log, procedure output, and online Help, and more. The following figure shows the SAS windowing environment.

Figure 1.1   SAS Windowing Environment
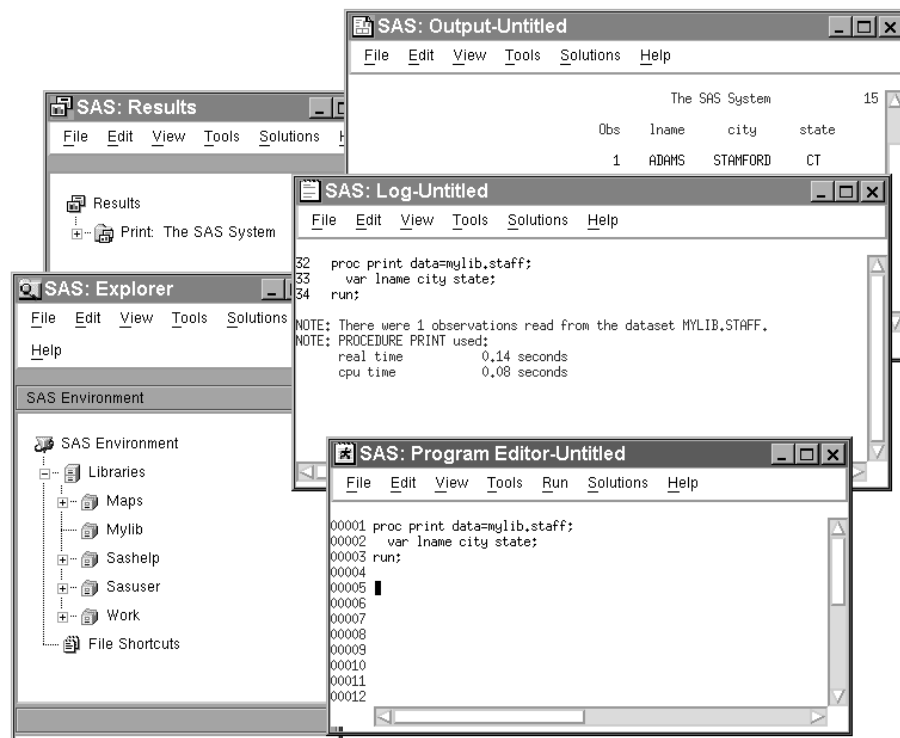
In the Explorer window, you can view and manage your SAS files, which are stored in libraries, and create shortcuts to non-SAS files. The Results window helps you navigate and manage output from SAS programs that you submit; you can view, save, and manage individual output items. You use the Progam Editor, Log, and Output windows to enter, edit, and submit SAS programs, view messages about your SAS session and programs that you submit, and browse output from programs that you submit. For more detailed information about the SAS windowing environment, see *Getting Started with the SAS System*.

Interactive Line Mode

In *interactive line mode*, you enter program statements in sequence in response to prompts from the SAS System. DATA and PROC steps execute when

- □ a RUN, QUIT, or a semicolon on a line by itself after lines of data are entered
- □ another DATA or PROC statement is entered
- □ the ENDSAS statement is encountered.

By default, the SAS log and output are displayed immediately following the program statements.

Noninteractive Mode

In *noninteractive mode*, SAS program statements are stored in an external file. The statements in the file execute immediately after you issue a SAS command referencing the file. Depending on your operating environment and the SAS system options that you use, the SAS log and output are either written to separate external files or displayed.

Operating Environment Information: Refer to the SAS documentation for your operating environment for information about how these files are named and where they are stored. △

Batch Mode

You can run SAS jobs in *batch mode* in operating environments that support batch or background execution. Place your SAS statements in a file and submit them for execution along with the control statements and system commands required at your site.

When you submit a SAS job in batch mode, one file is created to contain the SAS log for the job, and another is created to hold output that is produced in a PROC step or, when directed, output that is produced in a DATA step by a PUT statement.

Operating Environment Information: Refer to the SAS documentation for your operating environment for information about executing SAS jobs in batch mode. Also, see the documentation specific to your site for local requirements for running jobs in batch and for viewing output from batch jobs. △

Customizing Your SAS Session

Setting Default System Option Settings

You can use a *configuration file* to store system options with the settings that you want. When you invoke SAS, these settings are in effect. SAS system options determine how SAS initializes its interfaces with your computer hardware and the operating environment, how it reads and writes data, how output appears, and other global functions.

By placing SAS system options in a configuration file, you can avoid having to specify the options every time that you invoke SAS. For example, you can specify the NODATE system option in your configuration file to prevent the date from appearing at the top of each page of your output.

Operating Environment Information: See the SAS documentation for your operating environment for more information about the configuration file. In some operating environments, you can use both a system-wide and a user-specific configuration file. △

Executing Statements Automatically

To execute SAS statements automatically each time you invoke SAS, store them in an *autoexec file*. SAS executes the statements automatically after the system is initialized. You can activate this file by specifying the AUTOEXEC= system option.

Any SAS statement can be included in an autoexec file. For example, you can set report titles, footnotes, or create macros or macro variables automatically with an autoexec file.

Operating Environment Information: See the SAS documentation for your operating environment for information on how autoexec files should be set up so that they can be located by SAS. △

Customizing the SAS Windowing Environment

You can customize many aspects of the SAS windowing environment and store your settings for use in future sessions. With the SAS windowing environment, you can

- □ change the appearance and sorting order of items in the Explorer window
- □ customize the Explorer window by registering member, entry, and file types
- □ set up favorite folders
- □ customize the toolbar
- □ set fonts, colors, and preferences.

See the SAS online Help for more information and for additional ways to customize your SAS windowing environment.

How This Book is Organized

SAS System Concepts

In the *SAS System Concepts* section of this book, you learn about the basic elements of the SAS System that are the building blocks of SAS language: rules for words and names, variables, missing values, expressions, dates, times, and intervals, and each of the six SAS language elements – data set options, formats, functions, informats, statements, and system options.

SAS System Concepts also provides introductory information that helps you begin to use SAS, including information about the SAS log, SAS output, error processing, and debugging. Information about SAS processing prepares you to write SAS programs.

DATA Step Concepts

The *DATA Step Concepts* section provides detailed discussion and examples of how to write DATA step programs. This part of the book explains how to construct many types of programs and how SAS processes your programs. The discussion begins with an overview of DATA step processing and a walkthrough of a sample DATA step program.

Later sections cover more advanced topics, such as report writing, BY-group processing, array processing, and creating and executing stored compiled DATA step programs.

This part of the book also thoroughly examines SAS data sets and how to create and use them in your programs. Topics include reading raw data and reading, combining, and modifying SAS data sets.

SAS Files Concepts

The *SAS Files Concepts* section covers advanced topics that enable you to explore how individual pieces of the SAS System work. While you might not need much of this information to write effective SAS programs, you might find the information helpful for more advanced applications.

The section discusses and compares the elements that comprise the physical file structure that SAS uses, including data sets, data libraries, data files, data views, catalogs, engines, and external files. Advanced topics include the audit trail, integrity constraints, indexes, and file protection.