CHAPTER

26

SAS Data Libraries

Definition

A *SAS data library* is a collection of one or more SAS files that are recognized by the SAS System and that are referenced and stored as a unit. Each file is a member of the library.

The logical concept of a SAS data library remains constant, regardless of the operating environment. In any operating environment where SAS can be installed, the structure for organizing, locating and managing SAS files is the same.

At the operating environment level however, a SAS data library has different physical implementations. Most SAS data libraries implement the storage of files in a manner similar to the way the operating environment stores and accesses files.

For instance, in directory-based operating environments, a SAS data library is a group of SAS files that are stored in the same directory and accessed by the same engine. Other files can be stored in the directory, but only the files with file extensions assigned by SAS are recognized as part of the SAS data library. Under CMS, a SAS data library is a group of SAS files with the same filetype. Under OS/390, a SAS data library can be implemented as either a bound library in a traditional OS data set or as a directory under UNIX system services.

SAS files can be one of the following:

□ SAS data set

□ SAS catalog

□ SAS data view

□ stored compiled SAS program

□ SAS utility file

□ access descriptors

□ multi-dimensional database files such as MDDB, FDB, and DMDB files

□ item store files.

Figure 26.1 Types of Files in a SAS Data Library
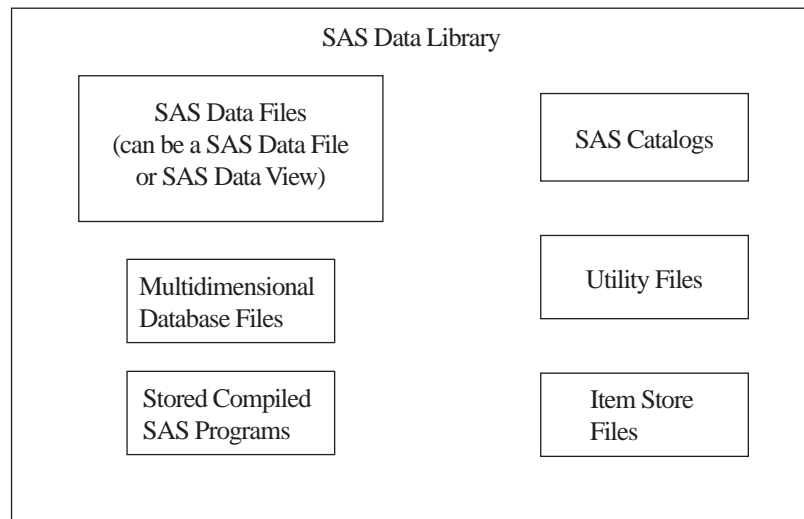
Each SAS file, in turn, stores information in smaller units that are characteristic of the *SAS file type*. For example, SAS data sets store information as variables and observations, while SAS catalogs store information in units called *entries*. SAS determines the type of a file from the context of the SAS program in which the file is created or specified; therefore, a library can contain files with the same name but with different member types.

SAS data libraries can contain files that you create, or they can be one of several special libraries that SAS provides for convenience, support, and customization capability such as the WORK library. SAS does not limit the number of SAS files you can store in a SAS data library.

For more information on SAS files, see the following sections:

□ "SAS Data Sets"

□ "SAS Data Files"

□ "SAS Data Views"

□ "SAS Catalogs"

□ "Stored Compiled SAS Programs"

□ "SAS File Utility Tasks."

Library Engines

Each SAS data library is associated with a SAS library engine. SAS library engines are software components that form the interface between the SAS System and the SAS data library. It is the SAS library engine that locates files in a SAS data library and renders the file contents to SAS in a form that it can recognize. Library engines perform such tasks as:

□ reading and writing data

□ listing the files in the library

□ deleting and renaming files.

SAS has a Multiple Engine Architecture that allows users to read to and write from files in different formats. Each SAS engine has specific processing characteristics, such as the ability to

□ process a SAS file generated by an older version of SAS

□ read database files created by other software programs

□ store and access files on disk or tape

□ determine how variables and observations are placed in a file

□ place data into memory from its physical location

□ transport SAS files between operating environments.

A user is generally not aware of the particular type of engine that is processing data for SAS at any given time. If the user issues a command that is not supported by the engine, an error message is displayed in the log.

More than one engine may be involved in processing a DATA step; for example, one engine may be used to input data, and another engine may be used to write observations to the output data set.

When needed, a user can select a specific engine to perform a task.

Engine name	Task
Remote	via SAS server
V6	for V6 libraries
V7	for V7 libraries
V8	for V8 libraries (default)
BMDP	for use with BMDP files
V6 tape	for V6 libraries on tape

Engine name	Task
XPORT	for reading data from a transport file

Library Names

Physical Names and Logical Names (Librefs)

Before you can use a SAS data library, you must show SAS where it is. SAS recognizes SAS data libraries based on either operating environment naming conventions, or on SAS naming conventions. There are three ways to define SAS data libraries.

☐ a physical location name that the operating environment recognizes

☐ a logical name (libref), that you assign using the SAS LIBNAME statement or function

☐ the SAS New Library window.

The physical name of the SAS data library is a name that identifies your SAS files to the operating environment. This physical library name must conform to the naming conventions of your operating environment. The physical name fully identifies the directory, or operating environment data set that contains the SAS data library.

The logical name, or *libref,* is the way you identify a group of files to SAS. A libref is a temporary name that you associate with the physical name of the SAS data library during each SAS job or session.

Librefs that you specify remain in effect until the SAS session ends, unless you create them in the SAS New Library window and check the "Enable at startup" box when you create the library. The New Library window is available in your Toolbar.

Assigning Librefs

Once the libref is assigned, you can read, create, or update files in a data library. You can also access files without using a libref. See "Accessing Permanent SAS Files Without a Libref." A libref is valid only for the current SAS session, unless it is assigned using the New Library window with the "Enable at startup" box checked. A libref can have a maximum length of eight characters. You can use the LIBREF function to verify that a libref has been assigned. Librefs can be referenced repeatedly within a SAS session. SAS does not limit the number of librefs you can assign during a session; however, your operating environment or site may set limitations. See the LIBREF function in *SAS Language Reference: Dictionary* for more information.

Librefs can be assigned with the following:

☐ the LIBNAME statement

☐ the LIBNAME function

☐ the SAS Explorer window

☐ operating environment commands.

Operating Environment Information: The rules for assigning and using librefs differ across operating environments. See the SAS documentation for your operating environment for more specific information. △

To assign a library name to your SAS data library using the physical name, you assign a libref to the library, followed by the physical location, enclosed in single quotation marks.

If running in batch mode, the library must exist before you can allocate or assign it. In interactive mode, you may be allowed to create it if it does not already exist.

Table 26.1   Syntax for Assigning a Libref

Operating Environment	Examples
DOS, Windows	`libname mylibref 'c:\root\mystuff\sasstuff\work' ;`
UNIX	`libname mylibref '/u/mystuff/sastuff/work';`
UNIX System Services under OS/390	`libname mylibref '/mystuff/sastuff/work';`
CMS	`libname mylibref '< filetype> filemode |* ';`
OS/390 Open Edition	`libname mylibref 'my.data_library';`
OS/390	`libname mylibref 'userid.mystuff.sastuff.work';`
VAX/ALPHA	`libname mylibref 'filename filetype filemode';`

Associating and Clearing Logical Names (Librefs) With the Libname Statement

You can assign or clear a physical name with a libref using the LIBNAME statement, or the LIBNAME function, which are described in the *SAS Language Reference: Dictionary*.

Operating Environment Information: For some operating environments, you can use operating environment commands to associate a libref with a SAS data library. When using operating environment commands to assign librefs to a SAS data library, the association may persist beyond the SAS session in which the libref was created. For some operating environments you can use only the LIBNAME statement or function. See the SAS documentation for your operating environment for more information on assigning librefs. △

The most common form of the LIBNAME statement is used in this example to associate the libref ANNUAL with the physical name of the SAS data library.

```
libname annual 'SAS-data-library';
```

If you use the LIBNAME statement to assign the libref, SAS clears (deassigns) the libref automatically at the end of each SAS session. If you want to clear the libref ANNUAL before the end of the session, you can issue the following form of the LIBNAME statement:

```
libname annual clear;
```

SAS also provides a New Library window to assign or clear librefs and SAS Explorer to view, add or delete SAS data libraries. You can select the New Library or the SAS Explorer icon from the Toolbar.

Reserved Librefs

SAS reserves a few names for special uses. You should not use SASHELP, SASUSER or SASWORK as librefs, except as intended. The purpose and content of these libraries are discussed later in this section.

Operating Environment Information: There are other librefs reserved for SAS under some operating environments. In addition, your operating environment may have reserved certain words that cannot be used as librefs. See the SAS documentation for your operating environment for more information. △

Library Concatenation

Definition

Concatenation is the logical combining of two or more libraries. Concatenation allows you to access the SAS data sets in several libraries with one libref.

You can concatenate two or more libraries by specifying their librefs or physical names in the LIBNAME statement or function.

Physical names must be enclosed in single or double quotation marks in a LIBNAME statement, otherwise SAS looks for a previously assigned libref with the same name.

In the following examples, summer, winter, spring fall and and annual are previously defined librefs:

```
libname annual (summer winter spring fall);

libname annual ('path1' 'path2' 'path3');

libname annual ('path' winter spring fall);

libname total (annual 'path');
```

How SAS Concatenates Library Members

When there are members of the same name in more than one library, the first occurrence of the member is used for input, update, output, renames and deletes.

This example contains three SAS data libraries, and each data library contains two SAS data files. The LIBNAME statement concatenates LIB1, LIB2 and LIB3:

```
LIBNAME fruit (lib1 lib2 lib3);
```

```
LIB1            LIB2            LIB3
----            ----            ----
apples          apples          oranges
pears           oranges         plums
```

The concatenated library FRUIT has the following members:

```
fruit
------
apples
pears
```

```
oranges
plums
```

Note: Output will always go to the first level. For example: "data fruit.oranges" writes to LIB1. △

Note that in this example, if the file APPLES in LIB1 was a different file from the APPLES in LIB2 and an update to APPLES was specified, it will only be updated in LIB1 because that is the first occurence of the member APPLES.

For complete documentation on library concatenation, see the LIBNAME statement or function in *SAS Procedures Guide*.

For more information on how specific operating environments handle concatenation, see the SAS documentation for your operating environment.

Rules for Library Concatenation

After you create a library concatenation, you can specify the libref in any context that accepts a simple (nonconcatenated) libref. These rules determine how SAS files (that is, members of SAS libraries) are located among the concatenated libraries:

- □ If you specify any options or an engine, they apply only to the libraries that you specified with the complete physical name, not to any library that you specified with a libref.
- □ When a SAS file is opened for input or update, the concatenated libraries are searched and the first occurrence of the specified file is used.
- □ When a SAS file is opened for output, it is created in the first library that is listed in the concatenation.

    Note: A new SAS file is created in the first library even if there is a file with the same name in another part of the concatenation. △

- □ When you delete or rename a SAS file, only the first occurrence of the file is affected.
- □ Any time a list of SAS files is displayed, only one occurrence of a filename is shown.

    Note: Even if the name occurs multiple times in the concatenation, only the first occurrence is shown. △

- □ A SAS file that is logically connected to another file (such as an index to a data set) is listed only if the parent file resides in that same library. For example, if library ONE contains A.DATA, and library TWO contains A.DATA and A.INDEX, only A.DATA from library ONE is listed. (See rule 4.)
- □ If any library in the concatenation is sequential, then all of the libraries are treated as sequential.
- □ The attributes of the first library that is specified determine the attributes of the concatenation. For example, if the first SAS data library that is listed is "read only," then the entire concatenated library is "read only."
- □ Once a libref has been assigned in a concatenation, any changes made to the libref will not affect the concatenation.

Permanent and Temporary Libraries

SAS data libraries are generally stored as permanent data libraries; however, SAS provides a temporary or scratch library where you can store files for the duration of a SAS session or job.

A *permanent SAS data library* is one that resides on the external storage medium of your computer and is not deleted when the SAS session terminates. Permanent SAS data libraries are stored until you delete them. The library is available for processing in subsequent SAS sessions. When working with files in a permanent SAS data library, you generally specify a libref as the first part of a two-level SAS filename. The libref tells SAS where to find or store the file.

Note: You can also skip using a libref and point directly to the file you want to use, using syntax that your operating system understands. An example of this in the Windows environment is

```
Data 'C:\root\sasfiles\myfile.ext';
```

Operating Environment Information: Files are specified differently in various operating environments. See the SAS documentation for your operating environment for more information. △

△

A *temporary SAS data library* is one that endures only for the current SAS session or job. SAS files that are created during the session or job are held in a special work space that may or may not be an external storage medium. This work space is generally assigned the default libref WORK. Files in the temporary WORK library can be used in any DATA step or SAS procedure during the SAS session, but they are typically not available for subsequent SAS sessions. Normally, you specify that data sets be stored in or retrieved from this library by specifying a one-level name. Files held in the WORK library are deleted at the end of the SAS session if it ends normally.

There are a number of SAS system options that enable you to customize the way you name and work with your permanent and temporary SAS data libraries. See the USER=, WORK=, WORKINIT, and WORKTERM system options in *SAS Language Reference: Dictionary* for more information.

SAS System Libraries

Four special SAS System supplied libraries provide convenience, support, and customization capability:

- □ WORK
- □ USER
- □ SASHELP
- □ SASUSER

WORK Library

Definition

The WORK library is the temporary, or scratch, library automatically defined by SAS at the beginning of each SAS session. The WORK library stores two types of temporary files: those that you create and those that are created internally by SAS as part of normal processing. Typically, the WORK library is deleted at the end of each SAS session if the session terminates normally.

Using the WORK Library

To store or retrieve SAS files in the WORK library, specify a one-level name in your SAS program statements. The libref WORK is automatically assigned to these files as a system default unless you have assigned the USER libref. The following examples contain valid names for SAS data sets stored in the WORK library:

□ `data test2;`

□ `data work.test2;`

□ `proc contents data=testdata;`

□ `proc contents data=work.testdata;`

□ `open ('testfunc');`

Operating Environment Information: The WORK library is implemented differently in various operating environments. See the SAS documentation for your operating environment for more information. △

Relation to the USER Library

While the WORK library is designed to hold temporary files used during the current SAS session, the USER library is designed to hold files after the SAS session is over. If you associate the libref USER with a SAS data library, use a one-level name to create and access files that are not deleted at the end of your SAS session. When SAS encounters a one-level filename, it looks first in the USER library, if it has been defined, and then it looks in WORK. If you wish to place a file in the USER library, so that it is not deleted after your SAS session is over, any single-level file goes there by default. At that point, if you want to create a temporary file in WORK, you must use a two-level name, such as `WORK.name`.

USER Library

Definition

The USER library allows you to read, create, and write to files in a SAS data library other than WORK without specifying a libref as part of the SAS filename. Once you associate the libref USER with a SAS data library, SAS stores any file with a one-level name in that library. Unlike the WORK library, files stored in this library are not deleted by SAS when the session terminates.

Ways to Assign the USER Libref

You can assign the USER libref using

□ the LIBNAME statement
□ the LIBNAME function
□ the USER= system option
□ an operating environment command.

In this example, the LIBNAME statement is used with a DATA step, which stores the data set REGION in a permanent SAS data library.

```
libname user 'SAS-data-library';
data region;
```

```
… more DATA step statements …
run;
```

In this example, the LIBNAME function assigns the USER libref:

```
data _null_;
   x=libname ('user', 'SAS-data-library');
run;
```

When assigning a libref using the USER= system option, you must first assign a libref to a SAS data library, then use the USER= system option to specify that library as the default for one-level names. In this example, the DATA step stores the data set PROCHLOR in the SAS data library TESTLIB.

```
libname testlib 'SAS-data-library';
options user=testlib;
data prochlor;
… more DATA step statements …
run;
```

Operating Environment Information: The methods and results of assigning the USER libref vary slightly from one operating environment to another. See the SAS documentation for your operating environment for more information. △

Relation to WORK Library

The USER libref overrides the default libref WORK for one-level names. When you refer to a file by a one-level name, SAS looks first for the libref USER. If USER is assigned to a SAS data library, files with one-level names are stored there. If you have not assigned the libref USER to a library, the files with one-level names are stored in the temporary library WORK. To refer to SAS files in the WORK library while the USER libref is assigned, you must specify a two-level name with WORK as the libref. Data files that SAS creates internally still go to the WORK library.

SASHELP Library

Each SAS site receives the SASHELP library, which contains a group of catalogs and other files containing information that is used to control various aspects of your SAS session. The defaults stored in this library are for everyone using SAS at your installation. Your personal settings are stored in the SASUSER library, which is discussed later in this section.

If SAS products other than base SAS are installed at your site, the SASHELP library contains catalogs that are used by those products. In many instances, the defaults in this library are tailored to your site by your SAS Software Representative. You can list the catalogs stored at your site by using one of the file management utilities discussed later in this section.

SASUSER Library

The SASUSER library contains SAS catalogs that enable you to tailor features of SAS for your needs. If the defaults in the SASHELP library are not suitable for your applications, you can modify them and store your personalized defaults in your

SASUSER library. For example, in base SAS software, you can store your own defaults for function key settings or window attributes in a personal profile catalog named SASUSER.PROFILE.

SAS assigns the SASUSER library during system initialization, according to the information supplied by the SASUSER system option.

A system option called RSASUSER allows the system administrator to control the mode of access to the SASUSER library at installations that have one SASUSER library for all users and that want to prevent users from modifying it.

Operating Environment Information: In most operating environments, the SASUSER data library is created if it does not already exist. However, the SASUSER library is implemented differently in various operating environments. See the SAS documentation for your operating environment for more information. △

Tools for Managing Libraries

SAS Utilities

The SAS utilities available for SAS file management are designed to enable you to work with more than one SAS file at a time, as long as the files belong to the same library. The advantage of learning and using SAS Explorer, functions, options and procedures is that they automatically copy, rename, or delete any index files or integrity constraints, audit trails, backups and generation data sets that are associated with your SAS data files. Another advantage is that SAS utility procedures work on any operating environment at any level.

There are several SAS window options, functions and procedures available for performing file management tasks. You can use the following features alone or in combination, depending on what works best for you. See "Choosing the Right Procedure" in the *SAS Procedures Guide* for detailed information on SAS utility procedures. The SAS windowing environment and how to use it for managing SAS files is discussed in online Help.

CATALOG procedure
  provides catalog management utilities with the COPY, CONTENTS, and APPEND procedures.

DATASETS procedure
  provides all library management functions for all member types except catalogs. If your site does not use the SAS Explorer, or if SAS executes in batch or interactive line mode, using this procedure can save you time and resources.

SAS Explorer
  includes windows that enable you to perform most file management tasks without submitting SAS program statements. Type LIBNAME, CATALOG or DIR in the Toolbar window to use SAS Explorer, or select the Explorer icon from the Toolbar menu.

DETAILS system option
  Sets the default display for file information when using the CONTENTS, or DATASETS procedure. When enabled, DETAILS provides additional information about files, depending on which procedure or window you use.

Library Directories

SAS Explorer and procedures enable you to obtain a list, or *directory*, of the members in a SAS data library. Each directory contains the name of each member and its member type. For the member type DATA, the directory indicates whether an index, audit trail, backup, or generation data set is associated with the data set. The directory also describes some attributes of the library, but the amount and nature of this information vary greatly with the operating environment.

Note: SAS data libraries can also contain various SAS utility files. These files are not listed in the library directory and are for internal processing. △

Accessing Permanent SAS Files without a Libref

SAS provides you with another method of accessing files in addition to creating a libref with the LIBNAME statement or using the New Library window. To use this method, enclose the filename, or the filename and path, in single quotatation marks.

For example, in a directory based system, if you want to create a data set named "mydata" in your default directory, that is, in the directory that you are running SAS in, you can write the following line of code:

```
data 'mydata';
```

SAS creates the data set and remembers its location for the duration of the SAS session.

If you omit the single quotation marks, SAS creates the data set "mydata" in the temporary WORK subdirectory, named WORK.mydata:

```
data mydata;
```

If you want to create a data set named "mydata" in a library other than the directory in which you are running SAS, enclose the entire path in quotation marks, following the naming conventions of your operating environment. For example, the following DATA step creates a data set named FOO in the directory C:\sasrun\mydata.

```
data 'c:\sasrun\mydata\foo';
```

This method of accessing files works on all operating environments and in most contexts where a libref.memname is accepted as a SAS dataset. Most dataset options can be specified with a quoted name.

You cannot use quoted names for:

- a SAS CATALOG
- MDDB, and FDB references
- contexts that do not accept a libref, such as the SELECT statement of PROC COPY and most PROC DATASETS statements
- PROC SQL
- DATA step, stored programs or views
- the Screen Control Language (SCL) open function.

Operating Environment	Examples
DOS, Windows	`data 'c:\root\mystuff\sasstuff\work\myfile' ;`
UNIX	`data '/u/root/mystuff/sastuff/work/myfile';`
UNIX System Services under OS/390	`data '/mystuff/sastuff/work/myfile';`

Operating Environment	Examples
CMS	`data '< filetype> filemode |* ';`
OS/390	`data '/mystuff/sastuff/work/myfile';`
VAX/ALPHA	`data 'filename filetype filemode';`

Operating Environment Commands

You can use operating environment commands to copy, rename, and delete the operating environment file or files that make up a SAS data library. However, to maintain the integrity of your files, you must know how the SAS data library model is implemented in your operating environment. For example, in some operating environments, SAS data sets and their associated indexes can be copied, deleted, or renamed as separate files. If you rename the file containing the SAS data set, but not its index, the data set will be marked as damaged.

CAUTION:

Using operating environment commands can damage files. You can avoid problems by always using SAS utilities to manage SAS files. △

Sequential Data Libraries

SAS provides a number of features and procedures for reading from and writing to files that are stored on sequential format devices, either disk or tape. Before you store SAS data libraries in sequential format, you should consider the following

☐ You cannot use random access methods with sequential SAS data sets.

☐ You can access only one of the SAS files in a sequential library, or only one of the SAS files on a tape, at any point in a SAS job.

For example, you cannot read two or more SAS data sets in the same library or on the same tape at the same time in a single DATA step. However, you can access

☐ two or more SAS files in different sequential libraries, or on different tapes at the same time, if there are enough tape drives available

☐ a SAS file during one DATA or PROC step, then access another SAS file in the same sequential library or on the same tape during a later DATA or PROC step.

Also, when you have more than one SAS data set on a tape or in a sequential library in the same DATA or PROC step, one SAS data set file may be opened during the compilation phase, and the additional SAS data sets are opened during the execution phase. For more information, see the SET statement OPEN= option in the *SAS Language Reference: Dictionary*

☐ For some operating environments, you can only read from or write to SAS data sets during a DATA or PROC step. However, you can always use the COPY procedure to transfer all members of a SAS data library to tape for storage and backup purposes.

☐ Considerations specific to your site can affect your use of tape. For example, it may be necessary to manually mount a tape before the SAS data libraries become available. Consult your operations staff if you are not familiar with using tape storage at your location.

Operating Environment Information: The details for storing and accessing Version 6 and Version 5 SAS files in sequential format vary with the operating environment. See the SAS documentation for your operating environment for further information. △