CHAPTER 5

Allocating External Files

Introduction **69**
Ways of Allocating External Files **69**
Using the FILENAME Statement or Function to Allocate External Files **70**
FILENAME Statement Syntax **71**
FILENAME Statement Examples **71**
Using the JCL DD Statement to Allocate External Files **72**
Using the TSO Allocate Command to Allocate External Files **73**
Allocating External Files on Tape **73**
Allocating Generation Data Sets **74**
Allocating a New Generation of a Generation Data Group **74**
Allocating an Existing Generation of a Generation Data Group **74**
Allocating Nonstandard External Files **75**
Allocating ISAM Files **75**
Allocating UNIX System Services Hierarchical File System Files **75**
Allocating PDSEs **75**
Concatenating External Files **75**
Displaying Information about External Files **76**
Deallocating External Files **76**

Introduction

External files are files whose format is determined by the operating environment rather than by the SAS System. External files include raw data files, JCL libraries, files that contain SAS programming statements, load libraries, and HFS files, which are part of UNIX System Services. In batch and noninteractive line modes, the SAS log and procedure output files are also external files.

Ways of Allocating External Files

To work with an external file in the SAS System, you must first allocate the file. File allocation is the process of identifying an external file to the SAS System. If you are allocating a new data set, such as a sequential file, partitioned data set (PDS), or partitioned data set extended (PDSE), you must specify that it is new and you must describe its structure and format. These actions are not required for new files in the Hierarchical File System (HFS).

You can allocate external files in the following ways:

□ If you plan to use an existing external file only once in your SAS program, then you can allocate it by simply specifying the physical file name in a SAS statement or command. For example, this INCLUDE command allocates an existing sequential data set and includes it into the PROGRAM EDITOR window:

```
include 'myid.report.data'
```

Similarly, this PROC PRINTTO statement allocates a new PDS member:

```
proc printto
   print='userid.output.data(rockport)'
   new;
```

□ If you plan to use the same external file several times in your SAS program, then use one of the following methods to allocate the file.

SAS FILENAME statement or function
: You can use these methods in all modes for most types of files. See "Using the FILENAME Statement or Function to Allocate External Files" on page 70 or "FILENAME" on page 190 for more information.

JCL DD statement
: You can use this method if you use OS/390 in batch mode. See "Using the JCL DD Statement to Allocate External Files" on page 72 for more information.

Note: Unlike the other two methods, if you use the JCL DD statement to allocate a file, there is no way to deallocate the file until the job ends. △

TSO ALLOCATE command
: You can use this method if you use OS/390 under TSO. See "Using the TSO Allocate Command to Allocate External Files" on page 73 for more information.

Each of these methods establishes a fileref or a DDname that you can subsequently use to refer to the file instead of specifying the data set name again. See "Referring to External Files" on page 78 for more information.

Using the FILENAME Statement or Function to Allocate External Files

The FILENAME statement and FILENAME function associate a SAS fileref (file reference name) with the operating environment's name for an external file. This is equivalent to allocating a physical file externally (using a JCL DD statement or a TSO ALLOCATE command) and assigning a fileref to it.

In interactive mode, if you issue a FILENAME statement or function or attempt to allocate a file with the FNAME window for a file that does not exist, and if you do not specify DISP=NEW, and if the file is not an HFS file, one of the following actions occurs:

□ If the SAS system option FILEPROMPT is in effect (the default), then a requestor window asks whether you want to create the external file. If you reply **Yes**, SAS creates the external file, using any attributes that you specified in the FILENAME statement. If you do not specify any attributes, SAS uses the values of the SAS system options FILEDEV=, FILEVOL=, FILEUNIT=, FILESPPRI=, and FILESPSEC=. See "System Options in the OS/390 Environment" on page 328 for information about these options.

□ If the SAS system option NOFILEPROMPT is in effect, an error message indicating that the file could not be allocated is written to the SAS log.

For further information on the FILENAME function, see "FILENAME" on page 190.

FILENAME Statement Syntax

This section provides only a brief overview of FILENAME statement syntax. For complete information about the FILENAME statement, see “FILENAME” on page 294.

The syntax of the FILENAME statement is

FILENAME *fileref* <*device-type* > ’*physical-filename*’ <*options* . . . >;

fileref

identifies the external file. The *fileref* must conform to the rules for DDnames. That is, it can consist of one to eight letters, numbers, or the national characters $, @, and #; the first character must be either a letter or a national character. You can subsequently use the fileref to refer to this file in your SAS session or batch job. (See “Referring to External Files” on page 78.)

device-type

enables you to route output to an output device, disk, or tape file by specifying device type. If *device-type* is not defined for a new file, its value is taken from the SAS system option FILEDEV=.

’*physical-filename*’ | (’*physical-filename-1*’ . . . ’*physical-filename-n*’) |
’*physical-filename* (*)’ | ’*physical-filename*(*beg**)’ | ’*physical-filename*(**end*)’

is the physical file name of the data set, enclosed in quotes (see “Specifying Physical Files” on page 13), or it can be a concatenation of physical file names. For a concatenation, enclose each data set name in quotes, and enclose the entire group of file-specifications in parentheses. The maximum number of data sets in a concatenation is 255.

For a concatenation of members in a PDS, an asterisk (*) can be used in a wildcard file specification. The syntax ’*physical-filename* (*)’ applies to all members of the PDS; (*beg**) applies to all members or files whose names begin with *beg*, and (**end*) applies to all files whose names end with *end*.

options

include standard options such as file disposition as well as options for SYSOUT data sets such as the destination for output and the number of copies desired. These options are described in detail in “FILENAME” on page 294. Generally, values for options may be specified either with or without quotes. However, values that contain special characters must be enclosed in quotes.

FILENAME Statement Examples

The following table provides examples of the FILENAME statement for OS/390.

Table 5.1 FILENAME Statement Examples

Type of File	New or Existing File?	Example

sequential	existing	`filename`
		`raw 'myid.raw.datax' disp=old;`

	new	`filename x`
		`'userid.newdata' disp=new`
		`   space=(trk,(5,1)) unit=3380 volume=xyzabc`
		`   recfm=fb lrecl=80 blksize=6160;`

partitioned	existing	`filename raw 'sas.raw.data(mem1)' disp=old;`

Type of File	New or Existing File?	Example

	new	`filename dogcat 'userid.sas8.physn(optwrk)'`
		`disp=new space=(trk,(1,3,1))`
		`volume=xxx111 recfm=fb lrecl=255`
		`blksize=6120 dsorg=po;`

partitioned extended	existing	`filename mypdse 'sas.test.pdse' disp=old;`

	new	`filename tpdse 'sas.test.pdse' dsntype=library`
		`space=(trk,(5,2,2)) lrecl=80    blksize=6160`
		`recfm=fb disp=(new, catlg) dsorg=po;`

UNIX System Services: HFS files	existing	`filename myhfs '/u/userid/myfile';`
	new	`filename myhfs '/u/userid/myfile';`

temporary	new	`filename nextone '&mytemp' disp=new`
		`space=(trk,(3)) lrecl=80 blksize=6160;`

tape	existing	`filename mytape 'prod.data' vol=myvol  unit=tape`
		`label=(1,SL);`

	new	`filename`
		`tranfile 'sas.cport.file'`
		`label=(1,SL)`
		`vol='042627' unit=cart blksize=8000`
		`disp=(new,keep);`

concatenated	existing	`filename`
		`concat12`
		`('prod.payroll.data' 'prod.trans(may)');`

wildcard	existing, in PDS	`filename`
		`wild`
		`'prod.payroll(d*)';`

terminal	n/a	`filename term1  '*';`
		or
		`filename term2 terminal;`

printer	n/a	`filename prnt unit=printer sysout=a;`
		or
		`filename prnt printer sysout=a;`

Using the JCL DD Statement to Allocate External Files

The syntax of the JCL DD statement is

//DDname DD DSN=*data-set-name,options*

options include options such as file disposition as well as options that describe the format of the file.

Here are some examples:

□ Allocating an existing sequential data set:

```
//BOOKS DD DSN=LIBRARY.CATALOG.DATA,DISP=SHR
```

□ Allocating a new sequential data set:

```
//REPORT  DD DSN=LIBRARY.REPORT.FEB08,DISP=(NEW,CATLG),
//           SPACE=(CYL,(1,1)),UNIT=SYSDA,
//           DCB=(LRECL=80,RECFM=FB,BLKSIZE=6160)
```

□ Concatenating sequential data sets:

```
//INPUT   DD DSN=LIBRARY.DATA.QTR1,DISP=SHR
//        DD DSN=LIBRARY.DATA.QTR2,DISP=SHR
//        DD DSN=LIBRARY.DATA.QTR3,DISP=SHR
//        DD DSN=LIBRARY.DATA.QTR4,DISP=SHR
```

For complete information about the JCL DD statement, see the appropriate JCL User's Guide and JCL Reference for your OS level.

Using the TSO Allocate Command to Allocate External Files

The syntax of the TSO ALLOCATE command is

ALLOC FILE(*DDname*) DA('*data-set-name* ') *options*

options include options such as file disposition as well as options that describe the format of the file.

Here are some examples:

- □ Allocating an existing member of a PDS:

```
alloc fi(in1) da('my.pds(mem1)') shr
```

- □ Allocating a new sequential data set:

```
alloc fi(report) da('library.report.feb08')
   new sp(1,1) cyl lrecl(80) recfm(f b)
   blksize(6160)
```

- □ Concatenating sequential data sets:

```
alloc fi(input) da('library.data.qtr1' 'library.data.qtr2'
   'library.data.qtr3' 'library.data.qtr4') shr
```

For complete information about the TSO ALLOCATE command, see the appropriate TSO reference for your OS level.

Allocating External Files on Tape

Tapes are used primarily in batch mode; in fact, some sites may restrict or prohibit tape mounts in interactive sessions. Because file allocation for external files on tape is done infrequently, the FILENAME statement or FILENAME function gives only limited support for parameters that are normally associated with data sets on tape. However, you can use the FILENAME statement or FILENAME function to allocate a cataloged tape file, provided that you specify the data set name and disposition (as you would normally do in a JCL DD statement). To allocate an uncataloged tape file, do the following:

- □ For a data set on an IBM standard-label tape (label type SL, the most common type), you must specify the data set name, UNIT= parameter, and volume serial number. You may also specify the label number and type, and the disposition, or you can allow default values to be used for these parameters. For example:

```
filename mytape 'prod.data' vol=myvol
   unit=tape label=(2,SL);
```

- □ For a data set on a nonlabeled tape (label type NL) you must supply the above information plus DCB information. (See “DCB Attribute Options” on page 300 for details.) For example:

```
filename tranfile 'sas.cport.data'
   disp=(new,keep) unit=tape vol=xvol
   label=(1,NL) recfm=fb
   lrecl=80 blksize=8000;
```

Allocating Generation Data Sets

A generation data set (or *generation*) is a version of an OS/390 data set that is stored as a member of a generation data group. These generations are supported by OS/390; they differ from the generation data sets supported by SAS. For detailed information about OS/390 generations, see your IBM documentation. For information on SAS generation data sets, see the *SAS Language Reference: Concepts*. See also “Allocating a Multivolume Generation Data Group” on page 45.

Both standard external files and SAS data libraries can be stored and managed as generation data groups. The following sections describe the various methods of allocating new and existing generations.

Allocating a New Generation of a Generation Data Group

To allocate a *new* generation of a generation data group, use one of the following methods:

□ In a JCL DD statement, you can specify either the relative form of the data set name or the absolute form.

Relative form:

```
//DD1  DD DSN=PROD.GDG(+1),DISP=(NEW,CATLG)
```

Absolute form:

```
//DD1  DD DSN=PROD.GDG.G0008V00,DISP=(NEW,CATLG)
```

□ In a SAS FILENAME statement or FILENAME function (for external files) or a TSO ALLOCATE command, you must specify the absolute form of the data set name.

FILENAME statement:

```
filename dd1 'prod.gdg.g0008v00'
   disp=(new,catlg);
```

TSO ALLOCATE command:

```
alloc fi(dd1) da('prod.gdg.g0008v00') new
```

Allocating an Existing Generation of a Generation Data Group

To access an existing generation of a generation data group, you can use either the relative form of the data set name or the absolute form in a FILENAME statement FILENAME function, JCL DD statement, or TSO ALLOCATE command.

□ Relative form:

FILENAME statement:

```
filename gdgds 'my.gdg.data(-1)';
```

JCL DD statement:

```
//DD1    DD DSN=PROD.GDG(-1),DISP=SHR
```

TSO ALLOCATE command:

```
alloc fi(dd1) da('prod.gdg(-1)') shr
```

□ Absolute form:

FILENAME statement:

```
filename gdgds 'my.gdg.data.g0008v01';
```

JCL DD statement:

```
//DD1    DD DSN=PROD.GDG.G0008V01,DISP=SHR
```

TSO ALLOCATE command:

```
alloc fi(dd1) da('prod.gdg.g0008v01')
   shr
```

Allocating Nonstandard External Files

Allocating ISAM Files

To allocate a new ISAM file, you must use either a JCL DD statement or the TSO ALLOCATE command; you cannot use the FILENAME statement or FILENAME function. However, you can use the FILENAME statement or function to allocate an existing ISAM file.

Allocating UNIX System Services Hierarchical File System Files

See “Accessing UNIX System Services Files” on page 93 for details.

Allocating PDSEs

To allocate a Partitioned Data Set Extended (PDSE), specify the appropriate options in the FILENAME statement or FILENAME function, as shown in the example in Table 5.1 on page 71.

See “Options That Specify SMS Keywords” on page 303 for definitions of SMS options.

You can use a PDSE wherever you can use a PDS, and you can write to multiple members in a PDSE at the same time.

Concatenating External Files

Multiple sequential data sets can be concatenated via JCL DD statements, a TSO ALLOCATE command, a FILENAME statement, or a FILENAME function. (When accessing concatenated files, performance is better when either of the first two methods is used.) See the examples in “Using the FILENAME Statement or Function to Allocate External Files” on page 70, “Using the JCL DD Statement to Allocate External Files” on page 72, and “Using the TSO Allocate Command to Allocate External Files” on page 73. Also see “Reading Concatenated Data Sets” on page 89.

Displaying Information about External Files

You can issue the FILENAME command from the command line to display the FILENAME window. This window lists all current SAS filerefs plus the name of the physical file to which each fileref has been assigned. Files that were allocated externally (with a JCL DD statement or with the TSO ALLOCATE command) are listed only after you have used them as filerefs in your SAS session. For more information about the FILENAME window, see *SAS Language: Reference*.

Under OS/390, three additional windows–FNAME, DSINFO, and MEMLIST–also provide information about external files. For information about these windows, see “Host-Specific Windows in the OS/390 Environment” on page 436.

Deallocating External Files

The method that you use to deallocate a file depends on which method you used to allocate it:

□ If you used the FILENAME statement or FILENAME function to allocate the file, include the CLEAR argument to deallocate it:

```
filename books clear;
```

Note: The CLEAR argument is optional. Specifying **`FILENAME`** ***`fileref`*****`;`** has the same effect. △

□ If you used the JCL DD statement to allocate the file, then the file is automatically deallocated when the job step ends. (There is no way to deallocate the file before the job step ends.)

□ If you used the TSO ALLOCATE command to allocate the file, then use the TSO FREE command:

```
free fi(books)
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OS/390 Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for the OS/390® Environment, Verison 8

1–58025–523–X

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

1st printing, November 1999

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