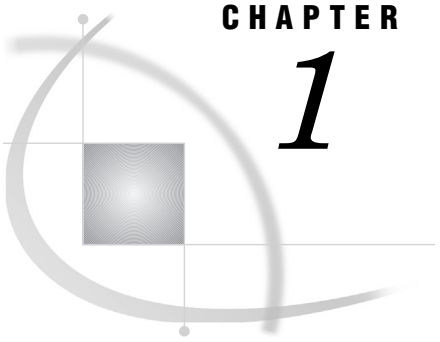

CHAPTER

1

Initializing and Configuring SAS Software

Invoking SAS in the OS/390 Environment

You can invoke SAS with any of the following methods:

- in interactive mode under TSO using the SAS CLIST
- in batch mode with the SAS cataloged procedure
- by logging on to SAS directly and bypassing the TSO terminal monitor program.

Invoking SAS under TSO: the SAS CLIST

To invoke SAS under TSO, you execute the SAS CLIST by typing a command (usually **SAS**) at the READY prompt. The SAS CLIST is an external file that contains TSO commands and control instructions.

At each site, the command that you use and the SAS CLIST itself might have been modified by your local SAS Support Consultant. Ask your consultant for site-specific information about the CLIST.

The SAS CLIST starts a SAS windowing environment session, an Explorer session, an interactive line mode session, or a noninteractive session, depending on the defaults that have been specified in the CLIST. To override the mode of running SAS that is specified in the CLIST, you use commands similar to those shown in Table 1.1 on page 5. (Again, the exact commands that you use may be site-specific.)

Table 1.1 Commands for Invoking SAS

Mode	How to Invoke	How to Terminate	Description
SAS windowing environment	`sas options('dms')`	**bye** or **endsas**	enables you to write and execute SAS programs and to view the SAS log and SAS procedure output in an interactive windowing environment. If this is the default at your site, then you can invoke it by entering **sas** with no options.
Explorer	`sas options('explorer')`	**bye** or **endsas**	enables you to manipulate SAS data and files visually, launch SAS applications, and access SAS windowing environment windows and Output Delivery System hierarchies.
interactive line mode	`sas options('nodms')`	**/*** or **endsas;** statement	prompts you to enter SAS statements at your terminal, one line at a time.
noninteractive mode	`sas input(''' my.sas.program''')`	n/a	executes under interactive OS/390, but it is called noninteractive because the program runs with no intervention from the terminal.

Invoking SAS in Batch Mode: the SAS Cataloged Procedure

To invoke SAS during a batch job, use a JCL EXEC statement that executes the SAS cataloged procedure. The SAS cataloged procedure invokes SAS. By specifying parameters in the JCL EXEC statement, you can modify the way in which SAS is invoked.

At each site, the JCL EXEC statement that you use and the cataloged procedure itself might have been modified by your local SAS Support Consultant. Ask your consultant for site-specific information.

Logging On to SAS Software Directly

OS/390 sites can choose to substitute SAS for the standard TSO terminal monitor program, enabling users to log on to SAS directly. If SAS comes up automatically when you log in, then your system may have already been set up to log on to SAS directly.

By automatically invoking SAS software or a SAS application when users log on, site administrators can insulate users from the TSO environment. Because SAS is running as its own terminal monitor program, TSO commands are not accessible to users. This reduces memory usage slightly.

This method of invoking SAS also provides the following advantages:

- Sites can restrict user access to the TSO environment.
- Novice users do not have to learn how to work in the TSO environment.

Your local SAS Support Consultant will find complete information about this method of invoking SAS in the installation instructions for SAS in the OS/390 environment.

Connecting to SAS under OS/390

Under OS/390, you can access or connect to a SAS session in any of the following ways:

3270 terminals
: You can use devices that support extended data streams as well as those that do not. See “Terminal Support in the OS/390 Environment” on page 558 for more information about terminal support.

terminal emulators
: Terminal emulators that you can use to access SAS on OS/390 include Attachmate Extra!, Hummingbird Host Explorer, and others.

SAS/CONNECT software
: SAS/CONNECT supports cooperative and distributed processing between OS/390 and Windows, UNIX, and OpenVMS Alpha. It supports several advanced communications protocols (including TCP/IP, APPC, TELNET, EHLLAPI, ASYNC, and 3270), enabling local clients who are running SAS to communicate with one or more SAS applications or programs that are running in remote environments. For more information, see *SAS/CONNECT User's Guide*.

SAS/SHARE software
: SAS/SHARE enables local and remote clients in a heterogeneous network to update SAS data concurrently. It also provides a low-overhead method for multiple remote clients to read local SAS data. For more information, see *SAS/SHARE User's Guide*.

SAS/SESSION software
: SAS/SESSION enables terminal users who are connected to the Customer Information Control System (CICS) to communicate with SAS software in an OS/390 environment. It uses the LU6.2 (APPC/MVS) protocol. Your local SAS Support Consultant will find more information about SAS/SESSION in the installation instructions for SAS software in the OS/390 environment.

Customizing Your SAS Session

Whether you are using interactive processing under TSO or batch processing, you might want to customize certain aspects of your SAS session. For example, you might want to change the line size or page size for your output, or you might want to see performance statistics for your SAS programs.

You can customize your SAS session in five ways:

- Under TSO, pass operands into the SAS CLIST that your site uses to invoke SAS. (See “Invoking SAS under TSO: the SAS CLIST” on page 4.) This method is usually used for one-time overrides of CLIST operands. Here is an example:

```
sas options('nocenter linesize=80')
```

- In batch mode, pass parameters into the SAS cataloged procedure that your site uses to invoke SAS. (See “Invoking SAS in Batch Mode: the SAS Cataloged Procedure” on page 5.) This method is usually used for one-time overrides of parameters in the cataloged procedure. Here is an example:

```
//MYJOB  EXEC SAS,
//   OPTIONS='NOCENTER, LINESIZE=80'
```

- Specify SAS system options in a user configuration file. (See "Configuration Files" on page 7.) This method is useful if you, as an individual user, always want to override the values of system options that are specified in your site's system configuration file. The following example uses a TSO command to specify a user configuration file:

  ```
  sas  config(''my.config.file'')
  ```

  This next example specifies a user configuration file using JCL:

  ```
  //MYJOB  EXEC SAS,
  //       CONFIG='MY.CONFIG.FILE'
  ```

- Execute SAS statements (such as OPTIONS, LIBNAME, and FILENAME statements) in an AUTOEXEC file. (See "Autoexec Files" on page 8.) This method is most useful for specifying options and allocating files that pertain to a particular SAS application.
- In interactive mode, specify a SASUSER library that contains a user profile catalog. (See "SASUSER Library" on page 9.)

See "Precedence for Option Specifications" on page 13 for information about the order of precedence for options specified using these methods.

Configuration Files

A *configuration file* contains SAS system options that are set automatically when you invoke SAS. SAS uses two types of configuration files:

- the system configuration file, which is used by all users at your site by default. Your local SAS Support Consultant maintains the system configuration file for your site.
- a user configuration file, which is generally used by an individual user or department.

Creating a User Configuration File

To create a user configuration file, use any text editor to write SAS system options into a physical file. The configuration file can be either a sequential data set or a member of a partitioned data set that contains 80-byte fixed-length records. When you allocate a system or user configuration file, you must specify LRECL=80 and RECFM=FB.

Whichever type of data set you choose, specify one or more system options on each line. If you specify more than one system option on a line, use either a blank or a comma to separate the options.

Some options can be thought of as on (enabled) or off (disabled). Specifying just the keyword enables the option; specifying the keyword prefixed with NO disables the option. For example, the configuration file might contain these option specifications:

```
NOCENTER
NOSTIMER
NOSTATS
```

All of these options are disabled.

Options that take a value must be specified in the following way:

option-name=value

For example, a configuration file might contain the following lines:

```
LINESIZE=80
PAGESIZE=60
```

Note: When you specify SAS system options in a configuration file, blank spaces are not permitted before or after an equal sign. Comment lines must start with an asterisk in column 1. △

A configuration file can contain any system option except the CONFIG= option. If CONFIG= appears in a configuration file, it is ignored; no error or warning message appears.

Specifying a User Configuration File

To tell SAS where to find your user configuration file, do the following:

- If you use the SAS CLIST to invoke SAS under TSO, use the CONFIG operand. For example:

```
sas config(''my.config.file'')
```

- If you use the SAS cataloged procedure to invoke SAS in batch mode, use the CONFIG= parameter. For example:

```
//S1  EXEC SAS,CONFIG='MY.CONFIG.FILE'
```

The user configuration file that you specify is executed along with the system configuration file that your installation uses. This happens because the SAS CLIST or the SAS cataloged procedure concatenates the file that you specified to the system configuration file.

Note: SAS system options that you specify in the user configuration file override system options that are specified in the system configuration file. △

Autoexec Files

Under OS/390, an *autoexec file* can be either a sequential data set or a member of a partitioned data set. Unlike configuration files, which contain SAS system options, an autoexec file contains SAS statements. These statements are executed immediately after SAS has been fully initialized and before any SAS input source statements have been processed. For example, an autoexec file could contain the following lines:

```
options fullstats pagesize=60 linesize=80;
libname mylib 'userid.my.lib';
dm 'clock';
```

The OPTIONS statement sets some SAS system options, the LIBNAME statement assigns a library, and the DM statement executes a command.

Note: Some SAS system options can be specified only when you invoke SAS. These system options cannot be specified in an OPTIONS statement; therefore, they cannot be specified in an autoexec file. See Table 19.4 on page 521 for information about SAS system options and where they can be specified. △

Displaying Autoexec Statements in the SAS Log

SAS statements that are submitted from an autoexec file usually are not displayed in the SAS log. However, if you specify the ECHOAUTO system option when you invoke SAS, then SAS writes (or "echoes") the autoexec statements to the SAS log as they are executed.

Using an Autoexec File under TSO

Under TSO, use the AUTOEXEC operand when you invoke SAS to tell SAS where to find your autoexec file. For example, the following command invokes SAS and tells SAS to use an autoexec file named MY.EXEC.FILE:

```
sas autoexec(''my.exec.file'')
```

Using an Autoexec File in Batch Mode

To specify an autoexec file in a batch job, use a JCL DD statement to assign the DDname SASEXEC to your autoexec file. This DD statement must follow the JCL EXEC statement that invokes the SAS cataloged procedure. For example, the following two lines of JCL can be used to accomplish the same results in a batch job as the previous example did under TSO:

```
//MYJOB    EXEC SAS
//SASEXEC  DD DSN=MY.EXEC.FILE,DISP=SHR
```

SASUSER Library

The SASUSER library contains SAS catalogs that enable you to customize certain features of SAS while your SAS session is running and to save these changes. For example, in base SAS software, any changes that you make to function key settings or to window attributes are stored in a catalog named SASUSER.PROFILE. The SASUSER library can also contain personal catalogs for other SAS software products. You can also store SAS data files, SAS data views, SAS programs, SAS/ACCESS descriptor files, and additional SAS catalogs in your SASUSER library.

When you use the SAS CLIST that is supplied by SAS to invoke SAS under TSO, the CLIST allocates a physical file to be used as the SASUSER library during your SAS session. The SASUSER library is normally used only in interactive processing; the SAS cataloged procedure, which invokes SAS in batch processing, does not allocate a SASUSER library.

In addition to storing function key settings and window attributes, the SASUSER.PROFILE catalog is used to store your DEFAULT.FORM. The DEFAULT.FORM is created by the FORM subsystem. It is used to control the default destination of all output that is generated by the PRINT command. (See "Using the PRINT Command and the FORM Subsystem" on page 123 and *SAS Language Reference: Dictionary* for information about the FORM subsystem.)

Note: If your SAS CLIST has been modified so that it does not create a SASUSER library, SAS creates a PROFILE catalog that is used to store profile information for use during a single SAS session. This catalog is placed in the WORK library and is deleted at the end of your session; it is not available in a subsequent SAS session. △

Creating Your Own SASUSER Libraries

By creating your own SASUSER libraries, you can customize SAS software to meet the requirements of a number of different types of jobs. For example, suppose you want to create a user profile for a particular type of task that requires a unique set of key definitions.

To create this user profile, you must first create a SAS data library that can be used as the SASUSER library. The easiest way to create this library is to start a Version 9 SAS session and then use a LIBNAME statement to create the library, as explained in "Allocating SAS Data Libraries Internally" on page 34. For example, to create a SAS

data library with a physical file name of ABC.MY.SASUSER, submit the following LIBNAME statement:

```
libname newlib 'abc.my.sasuser' disp=new;
```

Notice that a libref of NEWLIB was used in this example. SASUSER is a reserved libref and cannot be reassigned during a SAS session.

You can also use the TSO ALLOCATE command to create a physical file for use as your SASUSER library. By using the ALLOCATE command, you can avoid using the LIBNAME statement; however, you must be familiar with TSO commands and with DCB (data control block) attributes in order to use the ALLOCATE command effectively. Here is a typical ALLOCATE command for the SASUSER library that provides satisfactory performance at many sites:

```
alloc fi(newlib) da('abc.my.sasuser') new
      catalog space(80 20) dsorg(ps) recfm(f s)
      blksize(6144) reu
```

When you enter this ALLOCATE command from the READY prompt, a physical file named ABC.MY.SASUSER is created with the correct attributes for a SAS data library.

To use the new SAS data library as the SASUSER library, you must end your SAS session and start a second session. When you start a second session, you can use the SASUSER CLIST operand to specify ABC.MY.SASUSER as the SASUSER library.

Specifying Your Own SASUSER Library

After creating your own permanent SAS data library, designate that library as your SASUSER library. You can do this in either of the following ways:

- Use the SASUSER CLIST operand to specify the physical file name of your SAS data library. For example, if you had created a library with a name of ABC.MY.SASUSER, then you would use the following CLIST command to invoke SAS:

  ```
  sas sasuser(''abc.my.sasuser'')
  ```

  When you enter this command, the libref SASUSER is associated with the SAS data library whose physical file name is ABC.MY.SASUSER. Any profile changes that you make during your session are saved in the SAS catalog SASUSER.PROFILE, which is a member of the SASUSER library. These changes will be retained when you end your SAS session.

- Use the SASUSER= system option to specify the DDname that identifies your SAS data library. (See "SASUSER= System Option" on page 488.)

Both of these methods require that you identify the SAS data library when you invoke SAS; you cannot change the SASUSER library during a SAS session.

SAS System Options

SAS system options control many aspects of your SAS session, including output destinations, the efficiency of program execution, and the attributes of SAS files and data libraries.

After a system option is set, it affects all subsequent DATA and PROC steps in a process until it is specified again with a different value. For example, the CENTER|NOCENTER option affects all output from a process, regardless of the number of steps in the process.

Specifying or Changing System Option Settings

The default values for SAS system options are appropriate for many of your SAS programs. If you need to specify or change the value of a system option, you can do so in the following ways:

- Create a user configuration file to specify values for the SAS system options whose default values you want to override. See "Creating a User Configuration File" on page 7 for details.
- Under TSO, specify any SAS system option following the OPTIONS parameter in the SAS CLIST command:

  ```
  sas options('option-list')
  ```

  For options that can be on or off, just list the keyword that corresponds to the appropriate setting. For options that take a value, list the keyword identifying the option followed by an equal sign and the option value, as in the following example:

  ```
  sas options('nodate config=myconfig')
  ```

- In batch mode, specify any SAS system option in the EXEC SAS statement:

  ```
  // EXEC SAS,OPTIONS='option-list'
  ```

  For example:

  ```
  // EXEC SAS,OPTIONS='OPLIST LS=80 NOSTATS'
  ```

- Specify SAS system options in an OPTIONS statement in an autoexec file, which is executed when you invoke SAS, or in an OPTIONS statement at any point during a SAS session. Options specified in an OPTIONS statement apply to the process in which they are specified, and are reset for the duration of the SAS session or until you change them with another OPTIONS statement.

  For example:

  ```
  options nodate linesize=72;
  ```

  See Table 19.4 on page 521 to find out whether a particular option can be specified in the OPTIONS statement. For more information about autoexec files, see "Autoexec Files" on page 8. For more information about the OPTIONS statement, see *SAS Language Reference: Dictionary* and *Step-by-Step Programming with Base SAS Software*.
- Change SAS system options from within the OPTIONS window. On a command line, enter the keyword OPTIONS. The OPTIONS window appears. Place the cursor on any option setting and type over the existing value. The value will be saved for the duration of the SAS session only. Not all options are listed in the OPTIONS window. See "OPTIONS Window" on page 13 for more information.

Determining How an Option Was Set

Because of the relationship between some SAS system options, SAS may modify an option's value. This modification might change your results.

To determine how an option was set, enter the following code in the SAS Program Editor:

```
proc options option=option value; run;
```

After you submit this code, the SAS log will display the value that was set for the option and how the value was set. For example, the following log message is displayed when you enter

```
proc options option=CATCACHE value; run;
```

Output 1.1 Results of the OPTIONS Procedure for the CATCACHE Option

```
Option Value Information for SAS Option CATCACHE
      Option Value:  0
      Option Scope:  NoReb
      How option value was set:  Shipped Default
```

Options that are set by SAS will often say "Internal" in the **How option value was set** field.

Default Options Table and Restricted Options Table

Your local SAS Support Consultant may have created a default options table or a restricted options table. Information on creating and maintaining these tables is provided in the installation instructions for SAS software in the OS/390 environment.

The purpose of the default options table is to replace SAS system option defaults with values that are more appropriate for your site. You can change these new defaults in the same way that you can change the defaults provided with SAS software.

The purpose of the restricted options table is to control the values of invocation-only system options, which can be specified only when you invoke SAS. These values cannot be overridden. However, the restricted options table will accept specifications for any system option, including those that can be specified at any time during the SAS session. These specifications can be overridden at any time. To see when you can specify a value for a particular system option, refer to the "Summary Table of SAS System Options" on page 520.

You can determine where host options get their values by using the VALUE parameter of the OPTIONS procedure. For example, submit:

```
proc options host value;
run;
```

Then check the **How option value was set** field in the SAS log to determine if the value is the shipped default, or if the value was set in the default options table, or if the value was set in the SAS configuration file.

Contact your local SAS Support Consultant for more information.

Displaying System Option Settings

To display the current settings of SAS system options, use the OPTIONS procedure or the OPTIONS window.

Some options may seem to have default values even though the default value listed in Table 19.4 on page 521 is none. This happens when the option is set in a system configuration file, in the default options table, or in the restricted options table.

You can use the VALUE parameter of the OPTIONS procedure to see when an option's value was set.

OPTIONS Procedure

The OPTIONS procedure writes system options that are available under OS/390 to the SAS log. By default, the procedure lists one option per line with a brief explanation of what the option does. To list the options with no explanation, use the SHORT option:

```
proc options short;
run;
```

To list all the options in a certain category, use the GROUP= option:

```
proc options group=sort;
run;
```

Some options, such as system options that are specific to SAS/ACCESS interfaces or to the SAS interface to ISPF, are listed only if you specify the GROUP= option. See "OPTIONS Procedure" on page 317 for details.

OPTIONS Window

To display the OPTIONS window, enter **OPTIONS** on a command line. The OPTIONS window displays the settings of many SAS system options.

Precedence for Option Specifications

When the same option is set in more than one place, the order of precedence is as follows:

1. OPTIONS statement or OPTIONS window
2. restricted options table, if there is one
3. SAS invocation, including invocation by way of an EXEC SAS JCL statement (in batch) or by way of the SAS CLIST command (under TSO)
4. user configuration file, if there is one
5. system configuration file (as SAS software is initialized)
6. default options table, if there is one.

For example, options that you specify during your SAS session (using the OPTIONS statement or OPTIONS window) take precedence over options that you specified when you invoked SAS. Options that you specify with the SAS CLIST command take precedence over settings in the configuration file. The settings in the user configuration file take precedence over settings in the system configuration file and in the default options table.

Specifying Physical Files

Wherever you specify the name of a physical file internally (for example, in a SAS LIBNAME or FILENAME statement, in a LIBNAME or FILENAME function, in a DATA step, or in a SAS procedure), the name can be in any of these forms:

- a fully qualified data set name such as 'SAS.SAS9.AUTOEXEC' or 'MY.PDS(MEMBER)'.
- a partially qualified data set name such as '.CNTL'. SAS inserts the value of the SYSPREF= system option (which is usually *userid* by default) in front of the period. (See "SYSPREF= System Option" on page 511.) In the following example, an OPTIONS statement is used to assign a value of USER12.SAS9 to the SYSPREF= system option. When SAS executes the FILENAME statement, it interprets '.RAW.DATAX' as 'USER12.SAS9.RAW.DATAX'.

  ```
  options syspref=user12.sas9;
  filename raw2 '.raw.datax' disp=old;
  ```

- a temporary data set name such as '&MYTEMP'.
- a concatenated series of names or a wildcard name consisting of multiple UNIX System Services (USS) files or members of a partitioned data set (PDS, PDSE). See "Concatenating External Files" on page 86.

Note that names of physical files should be enclosed in quotation marks.

Specifying Physical Files with the INCLUDE Command

Here are examples of the INCLUDE command that illustrate the various ways you can specify physical files:

INCLUDE MYPGM
MYPGM is a fileref that was previously associated with an external file.

INCLUDE MYPGM(PGM1)
PGM1 is a member of the partitioned data set that is associated with the fileref MYPGM.

INCLUDE 'USERID.TEST.PGMS'
This is an example of a sequential data set name.

INCLUDE 'USERID.TEST.PGMS(AAA)'
This is an example of a data set name with a member specified.

INCLUDE '.TEST.MYPGM'
Assuming that the FILESYSTEM= system option is set to MVS, SAS prepends this data set name with the value of the SAS system option SYSPREF=, which defaults to the your system prefix. If FILESYSTEM=HFS, SAS looks into your default UNIX System Services directory for the "hidden" file .TEST.MYPGM.

INCLUDE 'HFS:/u/userid/mypgms/mypgm1.c'
This is an example of a path to a UNIX System Services (USS) file in the hierarchical file system, represented by a partially qualified path. SAS searches for the file in the default HFS directory for that user. If the FILESYSTEM= system option was set to HFS and if MYPGM was a standard OS/390 data set, the alternate syntax of MVS: would be required above (see "FILESYSTEM= System Option" on page 437).

INCLUDE 'pgms/mypgms/mypgm1.c'
This is another example of a relative path to a UNIX System Services file. Any file name containing a slash (/) is assumed to be in UNIX System Services, regardless of the value of the FILESYSTEM= system option.

Handling of Nonstandard Member Names

You can use the SAS system option FILEEXT= to specify how extensions in member names of partitioned data sets are to be handled. See "FILEEXT= System Option" on page 429 for more information.

SAS Software Files

Configuration files (described in "Configuration Files" on page 7) and SASUSER files (described in "SASUSER Library" on page 9) are only two of several SAS software files that are automatically identified to your session by either the SAS CLIST (under TSO) or the SAS cataloged procedure (in batch). This section describes several other SAS software files that are significant to SAS users under OS/390.

For brief descriptions of all the SAS software files that are frequently used by the SAS CLIST or by the SAS cataloged procedure, see Table 1.3 on page 21.

WORK Library

By default, the WORK library is a temporary SAS data library that contains temporary SAS data sets, utility files (created by some SAS procedures, such as PROC SORT and PROC TABULATE), your user profile, and other items that SAS uses in processing your current job. Anytime you assign a one-level name to a SAS data set, the data set is stored in the WORK library by default.

The WORK library is automatically defined by SAS software at the beginning of your SAS job or session, unless you invoke SAS under TSO and specify the GO operand. By default, the entire WORK library is deleted at the end of each SAS job or session.

The WORK library must exist on a disk device in Version 9 format so that it can be accessed by the V9 engine. (See "Using V9 Engines" on page 61 for information about the V9 engine.) Under OS/390, the physical file that is associated with the DDname WORK is allocated by the SAS CLIST or by the SAS cataloged procedure.

Space is the aspect of the WORK library that is most likely to require your consideration. Both the SAS cataloged procedure and the SAS CLIST include parameters that enable you to specify how much space to allocate to the work library. In the cataloged procedure and CLIST that are supplied by SAS, the space allocation for the WORK library is as follows:

```
SPACE=(6144,(500,200))
```

That is, the space is allocated in 6144-byte blocks, with a primary allocation of 500 blocks and a secondary allocation of 200 blocks. (Your installation may use different values; see the JCL from one of your SAS jobs to get a listing of the cataloged procedure that your SAS jobs use.) This space is enough for many SAS jobs. However, if you have many large temporary SAS data sets, or if you use a procedure that has many large utility files (for example, a PROC FREQ step with a complex TABLES statement that you run against a large SAS data set), you might run out of space in the WORK library. If you run out of space in batch mode, your PROC or DATA step terminates prematurely and issues a message similar to the one shown in the following output. In an interactive session, a dialog window asks you to specify what action to take.

Output 1.2 Insufficient WORK Space Message

```
ERROR: Insufficient space in file WORK.DATASET.DATA.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: SAS set option OBS=0 and will continue to check statements.
      This may cause NOTE: No observations in data set.
WARNING: The data set WORK.DATASET may be incomplete.  When this step
         was stopped there were 22360 observations and 4 variables.
ERROR: Errors printed on page 1.
```

Here are three possible solutions to this problem:

- Use a larger WORK library. (See "Increasing the Size of the WORK Library" on page 16.)
- Delete each temporary SAS data set as soon as you no longer need it. (See "Deleting Temporary SAS Data Sets" on page 16.)
- Direct the temporary SAS data sets to a different SAS data library so that data space in the WORK library is conserved for items that must be stored there. (See "Directing Temporary SAS Data Sets to the USER Library" on page 16.)

You can also combine these methods.

Increasing the Size of the WORK Library

Batch Mode Method

To increase the size of the WORK library in a batch job, include the WORK parameter in the EXEC statement in your JCL. The following SAS job allocates 1000 blocks of primary and 400 blocks of secondary space–twice as much as the default WORK allocations:

```
//HUGE JOB accounting-information
// EXEC SAS,WORK='1000,400'
//SYSIN DD *
SAS statements

/*
//
```

Interactive Mode Method

If you invoke SAS interactively, then include the WORK operand in the SAS CLIST command, as in the following example:

```
sas work('1000,400')
```

Deleting Temporary SAS Data Sets

Under OS/390, *temporary SAS data set* means a data set that is stored in a temporary SAS data library. That is, you cannot designate the data set itself as temporary, but the data set takes on the attribute of the library in which it is stored.

One simple way to conserve space in the WORK library is to delete each temporary SAS data set with a PROC DATASETS step after you no longer need it. However, there are two problems with this method.

- You can cause errors in a job by deleting a SAS data set before the job is finished with it.
- If you need several very large temporary SAS data sets in your job at the same time, you may run out of space before you reach a point at which you can delete any SAS data sets.

An alternative to deleting the temporary SAS data sets is to direct them to a different SAS data library, as described in the next section.

Directing Temporary SAS Data Sets to the USER Library

You can use the USER= system option to store temporary data sets in the USER library rather than in the WORK library. You can make the USER library as large as you need it to be.

Note: Utility data sets that are created by SAS procedures continue to be stored in the WORK library. However, any data sets that have one-level names and that are created by your SAS programs will be stored in the USER library. △

You can use a temporary or permanent physical file for the library, and you can put the library either on disk or on tape. The physical file can be either a Version 9, 8, 7, or 6 SAS data library. If it is a Version 6 SAS data library, then it provides support for data sets but not for catalogs. The following table summarizes differences between the WORK and USER libraries.

Table 1.2 Differences between the WORK and USER Libraries

Library	Type of Data Set	Storage Medium	Format
WORK	temporary	disk	V9
USER	temporary or permanent	disk or tape	V9, V8, V7, or V6

The following example illustrates the use of the USER= system option. The numbered lines of code are explained below.

```
   filename giant 'company.survey.tvdata';
   libname result 'my.tv.sasdata';
❶ libname temp '&tvtemp' space=(cyl,(6,2));
❷ options user=temp;
❸ data totalusa;
     infile giant;
     input home_id region income viewers cable;
     if home_id=. then delete;
   run;

❹ proc freq;
      tables region*income*viewers*cable
❺        / noprint out=result.freqdata;
   run;
```

1 The LIBNAME statement associates the libref TEMP with the temporary physical file &TVTEMP.

2 In the OPTIONS statement, the USER= system option designates the TEMP libref as the temporary SAS data library. Any data sets that have one-level names and that are created by your SAS program will be stored in this library.

3 A one-level name is used in the DATA statement. When the DATA step is processed, the SAS data set TEMP.TOTALUSA is created.

4 Because the large TOTALUSA data set was directed to the TEMP library, there is more space available in the WORK library for the utility files that the FREQ procedure requires.

5 The SAS data set FREQDATA contains the results of the FREQ procedure. A two-level name is used to store FREQDATA in the permanent SAS data library MY.TV.SASDATA.

SAS Log File

The SAS log file is a temporary physical file that has a DDname of SASLOG in both the SAS cataloged procedure and the SAS CLIST. In batch mode, the SAS cataloged procedure assigns default data control block (DCB) characteristics to this file as follows:

BLKSIZE=141

LRECL=137

RECFM=VBA

Under TSO, either interactively or noninteractively, the SASLOG file is routed to the terminal by default. In the windowing environment, the SAS log is directed to the Log window.

See “Types of SAS Output” on page 114 for more information about the SAS log and about how to route output in a batch job.

Changing the Contents of the SAS Log

The particular information that appears in the SAS log depends on the settings of several SAS system options. See “Collecting Performance Statistics” on page 216 for more information.

In addition, the following portable system options affect the contents of the SAS log:

CPUID
controls whether CPU information is printed at the beginning of the SAS log.

DETAILS
specifies whether to include additional information when files are listed in a SAS data library.

ECHOAUTO
controls whether the SAS source statements in the autoexec file are written (echoed) to the SAS log.

MLOGIC
controls whether macro trace information is written to the SAS log when macros are executed.

MPRINT
controls whether SAS statements that are generated by macros are displayed.

MSGLEVEL
controls the level of messages that are displayed.

NEWS=
specifies an external file that contains messages to be written to the SAS log when SAS software is initialized. Typically, the file contains information such as news items about the system.

NOTES
controls whether NOTES are printed in the log. NOTES is the default setting for all methods of running SAS. Do not specify NONOTES unless your SAS program is completely debugged.

OPLIST
specifies whether options given at SAS invocation are written to the SAS log.

PAGESIZE=
specifies the number of lines that compose a page of SAS output.

PRINTMSGLIST
controls whether extended lists of messages are printed.

SOURCE
controls whether SAS source statements are written to the log. NOSOURCE is the default setting for SAS interactive line mode; otherwise, SOURCE is the default.

SOURCE2
controls whether secondary source statements from files that are included by %INCLUDE statements are written to the SAS log.

SYMBOLGEN
controls whether the macro processor displays the results of resolving macro references.

Changing the Appearance of the SAS Log

The following portable system options are used to change the appearance of the SAS log:

DATE
controls whether the date and time, based on when the SAS job or session began, are written at the top of each page of the SAS log and of any print file that SAS software creates. Use NODATE to suppress printing of the date and time.

LINESIZE=
specifies the line size (printer line width) for the SAS log and the SAS procedure output file. LS= is an alias for this option. LINESIZE= values can range from 64 through 256.

NUMBER
controls whether the log pages are numbered. NUMBER is the default. Use the NONUMBER option to suppress page numbers.

OVP
controls whether lines in SAS output are overprinted.

SAS Procedure Output File

Whenever a SAS program executes a PROC step that produces printed output, SAS sends the output to the procedure output file. Under TSO, either interactively or noninteractively, the procedure output file is routed to the terminal by default. In the windowing environment, output is directed to the Output window.

In batch mode, the SAS procedure output file is identified in the cataloged procedure by the DDname SASLIST. Unless you specify otherwise, SAS writes most procedure output to this file. (A few procedures, such as the OPTIONS procedure, route output directly to the SAS log by default.) PUT statement output may also be directed to this file by a FILE statement that uses the fileref PRINT. (PRINT is a special fileref that can be specified in the FILE statement.)

The following DCB characteristics of the procedure output file are controlled by the cataloged procedure, typically with the following values:

BLKSIZE=264

LRECL=260

RECFM=VBA

The SAS procedure output file is often called the *print file*; however, any data set that contains carriage-control information (identified by a trailing A as part of the RECFM= specification) can be called a print file.

Changing the Appearance of Procedure Output

The following portable system options are used to change the appearance of procedure output:

CENTER
controls whether the printed results are centered or left-aligned on the procedure output page. CENTER is the default; NOCENTER specifies left alignment.

DATE
controls whether the date and time, based on when the SAS job or session began, are written at the top of each page of the SAS log and of any print file that SAS software creates. Use NODATE to suppress printing of the date and time.

LINESIZE=
specifies the line size (printer line width) for the SAS log and the SAS procedure output file. LS= is an alias for this option. LINESIZE= values can range from 64 through 256.

NUMBER
controls whether the page number is printed on the first title line of each SAS printed output page. NUMBER is the default. Use the NONUMBER option to suppress page numbers.

PAGENO=
specifies a beginning page number for the next page of output that SAS software produces.

PAGESIZE=
specifies how many lines to print on each page of SAS output. PS= is an alias for this option. In the windowing environment or in an interactive line mode session, the PAGESIZE= option defaults to the terminal screen size, if this information is available from the operating environment. PAGESIZE= values can range from 15 through 500.

Console Log File

The SAS console log file is a physical file that is automatically allocated at the start of SAS initialization. The name of the file is specified by the CONSOLELOG= system option. The console log file records log messages generated when the regular SAS log is either unavailable or is not yet allocated. You can control the appearance of the console log file with the LINESIZE= system option only. The SAS CLIST and catalogued procedures allocate this file using the DDname SASCLOG.

Parmcards File

The parmcards file is a temporary physical file that is identified by the DDname SASPARM. It is created automatically by the SAS cataloged procedure and by the SAS CLIST. SAS uses the parmcards file for internal processing. Lines that follow a PARMCARDS statement in a PROC step are first written to the parmcards file; then they are read into the procedure. The PARMCARDS statement is used in the BMDP and EXPLODE procedures.

Summary Table of SAS Software Files

Table 1.3 on page 21 lists all of the SAS software files that are frequently used in the SAS CLIST or in the SAS cataloged procedure. In the CLIST and cataloged procedure, logical names are associated with physical files. The logical names listed in the table are those that are used by the standard SAS CLIST or cataloged procedure. Your installation may have changed these names.

The system option column in the table lists the SAS system options that you can pass into the SAS CLIST (using the OPTIONS operand) or into the SAS cataloged procedure (using the OPTIONS parameter) when you invoke SAS. You can use these system options to change the defaults that were established by the CLIST or by the cataloged procedure. (See “Specifying or Changing System Option Settings” on page 11.)

Table 1.3 SAS Software Files

Default Logical Name	Purpose	System Option	CLIST Operands	Type of OS Data Set
CONFIG	system configuration file	CONFIG= *DDname*	DDCONFIG(*DDname*)	sequential data set or PDS member; must be FB, LRECL=80
Description: contains system options that are processed automatically when you invoke SAS. The system configuration file is usually maintained by your data center.				
CONFIG	user configuration file	CONFIG= *DDname*	CONFIG(*dsn*) DDCONFIG(*DDname*)	sequential data set or PDS member; must be FB, LRECL=80
Description: also contains system options that are processed automatically when you invoke SAS. Your user configuration file is concatenated to the system configuration file.				
LIBRARY	format library	n/a	n/a	SAS data library
Description: contains formats and informats.				
SAMPSIO	sample SAS data library	n/a	n/a	SAS data library
Description: is the SAS data library that is accessed by SAS programs in the sample library provided by SAS Institute.				
SAS*nnnnn*	command processor file	n/a	n/a	sequential data set or PDS member
Description: is used by the SASCP command in the SAS CLIST.				
SASAUTOS	system autocall library	n/a	MAUTS(*dsn*)	PDS
Description: contains source for SAS macros that were written by your data center or provided by SAS Institute.				
SASAUTOS	user autocall library	SASAUTOS= *specification**	SASAUTOS(*dsn*) DDSASAUT(*DDname*)	PDS
Description: contains a user-defined autocall library to which the system autocall library is concatenated.				
SASCLOG	console log	CONSOLELOG=*DDname*	n/a	sequential data set or PDS member
Description: SAS console log file.				
SASEXEC	autoexec file	AUTOEXEC= *DDname*	AUTOEXEC(*dsn*) DDAUTOEX(*DDname*)	sequential data set or PDS member
Description: contains statements that are executed automatically when you invoke SAS.				
SASHELP	HELP library	SASHELP= *DDname*	SASHELP(*dsn*) DDSASHLP(*DDname*)	SAS data library
Description: contains system default catalogs and Help system information.				
SASLIB	format library (V5)	SASLIB= *DDname*	n/a	load library
Description: a load library that contains user-written procedures and functions or Version 5 formats and informats. It is searched before the SAS software load library.				
SASLIST	procedure output file	PRINT= *DDname*	PRINT(*dsn*) DDPRINT(*DDname*)	sequential data set or PDS member

Default Logical Name	Purpose	System Option	CLIST Operands	Type of OS Data Set
Description: contains SAS procedure output.				
SASLOG	log file	LOG= *DDname*	LOG(*dsn*) DDLOG(*DDname*)	sequential data set or PDS member
Description: SAS log file.				
SASMSG	system message file	SASMSG= *DDname*	SASMSG(*dsn*) DDSASMSG(*DDname*)	PDS
Description: contains SAS software messages.				
SASPARM	parmcards file	PARMCARD= *DDname*	PARMCARD(*size*) DDPARMCD(*DDname*)	sequential data set or PDS member
Description: a temporary data set that is used by some procedures. The PARMCARD= system option assigns a DDname to the parmcards file; the PARMCARD CLIST operand specifies the file size. You can use the DDPARMCD operand to specify an alternate name for the parmcards file via the CLIST.				
SASSNAP	SNAP dump file	n/a	n/a	sequential data set or PDS member
Description: SNAP output from dump taken during abend recovery.				
SASSWK*nn*	sort work files	DYNALLOC SORTWKDD= SORTWKNO=	n/a	sequential
Description: temporary files that are used by the host sort utility when sorting large amounts of data.				
SASUSER	SASUSER library	SASUSER= *DDname*	SASUSER(*dsn*) DDSASUSR(*DDname*)	SAS data library
Description: contains the user profile catalog and other personal catalogs.				
STEPLIB	STEPLIB library	n/a	LOAD(*dsn*) SASLOAD(*dsn*)	load library
Description: a load library that contains SAS procedure and user-written load modules. (Allocate with a STEPLIB DD statement in a batch job.)				
SYSIN	primary input file	SYSIN= *DDname*	INPUT(*dsn*) DDSYSIN(*DDname*)	sequential data set or PDS member
Description: contains SAS statements. The primary input file can be specified with the INPUT operand under TSO, or allocated with a DD statement in a batch job.				
USER	USER library	USER= *DDname* | *dsn*	n/a	SAS data library
Description: specifies a SAS data library in which to store SAS data sets that have one-level names (instead of storing them in the WORK library).				
WORK	WORK library	WORK= *DDname*	DDWORK(*DDname*)	SAS data library
Description: contains temporary SAS files that are created by SAS software during your session.				

* SASAUTOS: *specification* can be a fileref, a partitioned data set name enclosed in quotation marks, or a series of file specifications enclosed in parentheses.

Transporting SAS Data Sets between Operating Environments

SAS supports three ways of transporting SAS data sets between OS/390 and other SAS operating environments: the XPORT engine, the CPORT and CIMPORT procedures, and SAS/CONNECT software, which is licensed separately. The process of moving a SAS file to or from OS/390 with the XPORT engine or with the CPORT and CIMPORT procedures involves three general steps:

1. Convert the SAS file to the intermediate form known as *transport format*.
2. Physically move the transport format file to the other operating environment.
3. Convert the transport format file into a normal, fully functional SAS file, in the format required by the other operating environment.

For further information on the XPORT engine and on the CPORT and CIMPORT procedures, including limited restrictions, refer to *Moving and Accessing SAS Files*.

SAS/CONNECT software allows you to move files between operating environments without using the intermediate transport format. For further information on SAS/CONNECT, including limited restrictions, refer to *SAS/CONNECT User's Guide*.

Accessing SAS Files in Other Operating Environments

SAS supports read-only cross-environment data access (CEDA) for certain types of SAS files created in the format of SAS Version 7 or later. CEDA allows you to read files in other operating environments as if those files were stored under OS/390. For further information on CEDA, refer to *Moving and Accessing SAS Files*.

Utilizing Input/Output Features

Version 5 and 6 data sets generally need to be moved to Version 9 if you want to take advantage of the I/O features introduced in SAS Version 9 and Version 8. For example, if you wanted to add integrity constraints to a Version 6 data set, you would first have to move that data set to Version 9. For information on upgrading your data sets, refer to *Moving and Accessing SAS Files*. For information on I/O features introduced in SAS Version 9, refer to the *SAS Language Reference: Dictionary*.

Reserved OS/390 DDnames

In addition to the logical names shown in Table 1.3 on page 21, which have a special meaning to SAS, you should be aware of the following reserved DDnames, which have a special meaning to the operating environment:

JOBCAT
: specifies a private catalog that the operating environment is to use instead of the system catalog for the duration of the job (including jobs with more than one job step).

JOBLIB
: performs the same function as STEPLIB (described in Table 1.3 on page 21) except that it can be used in a job that has more than one job step.

PROCLIB
specifies a private library of cataloged procedures to be searched before the system library of cataloged procedures is searched. See your local SAS Support Consultant for information about whether the PROCLIB DDname convention is used at your facility.

SORTLIB
is used by some host sort utilities.

SORTMSG
is used by some host sort utilities to print messages.

SORTWK*nn*
specifies sort work data sets for the host sort utility. If allocated, this will be used instead of the SASSWK*nn* data sets.

STEPCAT
specifies a private catalog that the operating environment is to use instead of the system catalog for the current job step.

SYSABEND
in the event of an abnormal job termination, SYSABEND specifies a data set that receives a medium-sized dump that consists of user-allocated storage and modules, system storage related to current tasks and open files, and system and programs related to the terminated job. See also SYSMDUMP and SYSUDUMP below.

SYSHELP
is used by TSO HELP libraries (not the SAS HELP facility).

SYSLIB
is used by some IBM system utility programs.

SYSMDUMP
in the event of an abnormal job termination, SYSMDUMP specifies a data set that receives a system dump in IPCS format. The contents of the dump are determined by OS/390 installation options, though SYSMDUMP generally includes all user-allocated storage, all system-allocated storage used to control job execution, and all program modules (system modules and user programs) that were in use at the time the dump was taken.

SYSOUT
is used by some utility programs to identify an output data set.

SYSPRINT
is used by some utility programs to identify a data set for listings and messages that may be sent to the printer.

SYSUADS
is used by some TSO commands that may be invoked under SAS software.

SYSUDUMP
in the event of an abnormal job termination, SYSUDUMP specifies a data set that receives a “short” system dump that consists of user-allocated storage and modules and system storage related to current tasks and open files. See also SYSABEND and SYSMDUMP above.

SYS*nnnnn*
is reserved for internal use (for dynamic allocation) by the operating environment.

Support for SAS Software

Support for SAS software is shared by SAS and your installation or site. SAS provides maintenance for the software; the SAS Installation Coordinator, SAS Support Consultant, and the SAS Training Coordinator for your site are responsible for providing you with direct user support.

- The SAS Installation Coordinator receives all shipments and correspondence and distributes them to the appropriate personnel at your site.
- The SAS Support Consultant is a knowledgeable SAS user who supports the other SAS users at your site. The SAS Technical Support Division is available to assist your SAS Support Consultant with problems that you encounter.
- The SAS Training Coordinator works with the SAS Education Division to arrange training classes for SAS users.

Working with Your SAS Support Consultant

At your site, one or more SAS Support Consultants have been designated as the first point of contact for SAS users who need help resolving problems.

If the SAS Support Consultant is unable to resolve your problem, then he or she will contact the SAS Technical Support Division for you. In order to provide the most efficient service possible, the company asks that you do not contact Technical Support directly.

SAS Technical Support

The SAS Technical Support Division can assist with suspected internal errors in SAS software and with possible system incompatibilities. It can also help answer questions about SAS statement syntax, general logic problems, and procedures and their output. However, the Technical Support Division cannot assist with special-interest applications, with writing user programs, or with teaching new users. It is also unable to provide support for general statistical methodology or for the design of experiments.

Generating a System Dump for SAS Technical Support

Follow these steps to generate a system dump that can be interpreted by SAS Technical Support:

1. Disable ABEND-AID or any other dump formatting system before generating the dump.
2. Allocate the DDname SYSUDUMP to an appropriate SYSOUT class, using either a TSO ALLOC command or the SUSUDUMP DD card in the cataloged procedure provided by SAS software.
3. Specify the following options at SAS invocation: NOSTAE, DUMPPROL, SOURCE, SOURCE2, NOTES, MPRINT, and SYMBOLGEN.

When the dump is produced, copy the dump file and any other output to a standard label machine-readable medium for shipment to SAS.

Solving Problems under OS/390

As you use SAS software under OS/390, you might encounter many different kinds of problems. Problems might occur within the context of your SAS program, or they might be with some component of the operating environment or with computer resources rather than with SAS software. For example, problems might be related to job control language or to a TSO command.

Problems Associated with the OS/390 Operating Environment

If a problem is detected by the operating environment, it sends messages to the job log or to the terminal screen (not the SAS log). In this case, you might need to consult an appropriate IBM manual or your on-site systems staff to determine the problem and the solution.

Most error messages indicate which part of the operating environment is detecting the problem. Here are some of the most common message groups, along with the operating environment component or utility that issues them:

CSV*xxxx*
: OS/390 load module management routines

ICE*xxxxx*
: IBM sort utility

ICH*xxxx*
: RACF system-security component of OS/390

IDC*xxxxx*
: catalog-management component of OS/390

IEC*xxxxx*
: OS/390 data-management routines

IKJ*xxxx*
: TSO terminal monitor program (TMP)

WER*xxxxx*
: SYNCSORT program

Consult the appropriate system manual to determine the source of the problem.

Solving Problems within SAS Software

Several resources are available to help you if you determine that your problem is within SAS software. These resources are discussed in the following sections.

Examining the SAS Log

The primary source of information for solving problems that occur within SAS software is the SAS log. The log lists the SAS source statements along with notes about each step, warning messages, and error messages. Errors are flagged in the code, and a numbered error message is printed in the log. It is often easy to find the incorrect step or statement just by glancing at the SAS log.

Checking the Condition Code

Upon exit, SAS returns a condition code to the operating environment that indicates its completion status. The condition code is translated to a return code that is

meaningful to the operating enviornment. SAS issues the condition codes in the following table:

Table 1.4 OS/390 Condition Codes

Return Code	Meaning
0	Successful completion
4	WARNING message(s) issued
8	Non-fatal ERROR message(s) issued
12	Fatal ERROR message(s) issued
16	ABORT; executed
20	ABORT RETURN; executed
ABND	ABORT ABEND; executed

Using SAS Online Help

Help is available through the SAS online help facility. To obtain host-specific help, execute the PMENU command as necessary to display SAS menus, then select

Help ► SAS System Help ► Main TOC ► Using SAS Software in OS/390

Then select topics of interest at increasing levels of detail.

Issue the KEYS command to determine the function keys used to page up, down, left, and right through help pages, and to move backward and forward between help topics.

Using User-Defined Help

Your site may provide user-defined help that provides site-specific information via the standard SAS help browser. To access user-defined help via the SAS help browser, you need to allocate a user-defined help library at SAS invocation.

The user-defined help library contains help information in the form of one or more *itemstores*, which utilize a file format that allows SAS to treat the itemstore as a file system within a file. Each itemstore can contain directories, subdirectories, and individual help topics. For information on loading user-defined help into itemstores, refer to "ITEMS Procedure" on page 314.

Help for SAS software is contained in itemstores. SAS automatically allocates libraries for SAS software help at SAS invocation. To invoke SAS so that it recognizes user-defined help, follow these steps:

1. In an autoexec file, allocate the SAS library that will contain the user-defined itemstore(s) using the LIBNAME statement. For example, if the libref is to be MYHELP and the itemstore is named APPL.HELP.DATA, the LIBNAME statement in the SAS invocation would be

   ```
   libname myhelp 'appl.help.data' disp=shr;
   ```

   See "Autoexec Files" on page 8 and "LIBNAME Statement" on page 388 for details.

2. Concatenate your itemstore(s) to the SAS help itemstore named by the HELPLOC= system option at SAS invocation. For example, if the libref for your user-defined help was MYHELP, and if the itemstore in the libref was named PRGAHELP, then the HELPLOC= specification in the SAS invocation would be as follows:

   ```
   helploc='myhelp.prgahelp'
   ```

See “HELPLOC= System Option” on page 446 for details on the HELPLOC= system option.

User-defined help cannot be added to the SAS help itemstore because most users have read-only access to the SAS help library.

After SAS has been invoked so that it can recognize user-defined help, you can access that help with the standard SAS help browser by issuing the HELP command and specifying the appropriate universal resource locator (URL). For example, if the help topic that you want to display is named DIRAHLP1.HTM, and if that help topic is contained in an itemstore directory named PRGADIRA, the HELP command would be as follows:

```
help helploc://prgadira/dirahlp1.htm
```

See the next section for information on developing user-defined help for the SAS help browser.

Developing User-Defined Help

You can develop help for your site or for your SAS programs that can be displayed in the standard SAS help browser. To ensure that your user-defined help will be displayed as it is written, use only the subset of tags from HTML that are supported on the SAS help browser. Help information in tags that are not supported by the SAS help browser might be ignored by the SAS help browser.

The following table describes the HTML tags supported by the SAS help browser. In short, the TABLE tag is the only frequently used tag that is not supported at this time. To add tables to your help, use the PRE tag and format the text manually using blank spaces, vertical bars, dashes, and underscores as needed.

Table 1.5 HTML Tags Supported by the SAS Help Browser

Tag Type	Tag Names	Description
heading	H1, H2, H3, H4, H5, H6	for hierarchical section headings
paragraph	P	for text in the body of a help file
list	UL, OL, DIR, MENU	for unordered (bullet) lists, ordered (numbered) lists, directory (unordered, no bullets) lists, and menu (unordered) lists
definition list	DL, DT, DD	for definition lists, titles of items, and definitions of items
preformatted text	PRE, XMP, LISTING	for tables, which must be manually formatted with blank spaces
font specification	I, B, U	for italic, bold, and underlined text
phrase	EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE	for emphasis, strong emphasis, definitions, code examples, code samples, keyboard key names, variables, citations
link	A, LINK	for anchors and the links that reference those anchors
document	TITLE, BASE, HEAD, HTML	for titles in the browser, base URLs, heading sections at the top of a page

For information on the options available for these tags, see any reference for the version of HTML supported by your browser.

For information on loading your help into itemstores, see "ITEMS Procedure" on page 314.

Using the SAS OnlineDoc CD-ROM

The CD-ROM that is supplied with SAS software contains most of the documentation for base SAS, including *SAS Language Reference: Dictionary* and other titles. If you encounter a problem that cannot be solved based on the information provided in the SAS log or in SAS online help, load the CD-ROM disk into a CD-ROM reader and browse through the contents of the books contained therein.

DATA Step Debugger

The DATA step debugger is an interactive tool that helps you find logic errors, and sometimes data errors, in SAS DATA steps. By issuing commands, you can execute DATA step statements one by one or in groups, pausing at any point to display the resulting variable values in a window. You can also bypass the execution of one or more statements. For further information on the DATA step debugger, see the *SAS Language Reference: Dictionary*.

Using SAS Statements, Procedures, and System Options to Identify Problems

If you are having a problem with the logic of your program, there might be no error messages or warning messages to help you. You might not get the results or output that you expect. Using PUT statements to write messages to the SAS log or to dump the values of all or some of your variables might help. Using PUT statements enables you to follow the flow of the problem and to see what is going on at strategic places in your program.

Some problems might be data related; these can be difficult to trace. Notes that appear in the SAS log following the step that reads and manipulates the data might be very helpful. These notes provide information such as the number of variables and observations that were created. You can also use the CONTENTS and PRINT procedures to look at the data definitions as SAS recorded them or to actually look at all or parts of the data in question.

SAS system options can also assist with problem resolution. Refer to the *SAS Language Reference: Dictionary* for details on the following system options and others that affect problem resolution:

MLOGIC
: controls whether SAS traces execution of the macro language processor.

MPRINT
: displays SAS statements that are generated by macro execution.

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

SOURCE2
: writes secondary source statements from included files to the SAS log.

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

Host-System Subgroup Error Messages

See “Messages from the SASCP Command Processor” on page 591 for brief explanations of many of the host-system subgroup error messages that you might encounter during a SAS session.