CHAPTER

# 19

# Macro Facility



## SAS Macro Facility under OpenVMS

In general, the SAS macro language is portable across operating environments. This section discusses those components of the macro facility that have details that are specific to OpenVMS. For more information, see the SAS online Help system for the macro facility, *SAS Macro Language: Reference*, or *SAS Macro Language: Reference*.

## Automatic Macro Variables

The following automatic macro variables have aspects that are specific to the OpenVMS operating environment:

SYSCC

specifies a character string that can be passed to SAS programs. By default, the value of SYSPARM is in uppercase in the OpenVMS operating environment. To preserve the case of the string, enclose it in double quotation marks.

SYSDEVIC

gives the name of the current graphics device.

SYSENV

is provided for compatibility with the SAS System running on other operating environments, but it is not relevant in the OpenVMS operating environment. In the OpenVMS environment, its value is always **FORE**.

SYSJOBID

lists the OpenVMS process ID (PID) of the process that is running the SAS System, (for example, **27A0D1D2**).

SYSPARM

specifies a character string that can be passed to SAS programs. By default, the value of SYSPARM is in uppercase in the OpenVMS operating environment. To preserve the case of the string, enclose it in double quotation marks.

SYSRC

holds the OpenVMS status of DCL commands that were issued during your SAS session. The variable holds a character string that is the text form of the decimal value of the OpenVMS command status. For example, consider the following statements:

```
x 'dirf'; /* an invalid OpenVMS command */
%put This OpenVMS status is &sysrc;

x 'dir'; /* a valid OpenVMS command */
%put The corrected OpenVMS status is &sysrc;
```

When these statements are issued, the following lines are written to the SAS log:

```
This OpenVMS status is 229520
The corrected OpenVMS status is 0
```

SYSSCP

in the OpenVMS Alpha operating environment, SYSSCP returns the value **VMS_AXP**. In the OpenVMS VAX operating environment, SYSSCP returns the value **VMS**.

SYSSCPL

returns the value **OpenVMS** for both the OpenVMS Alpha and VAX operating environments.

SYSSESID

returns the client name of an application. The client name of a SAS session consists of either the value of the **xresources=''name''** or the word **SAS** plus the SAS session ID. **SAS** is the default. The SAS session ID is incremented once for each concurrent session that is started. For example, if you start a second SAS session without ending the first session, then the default client name for the second session is **SAS2**.

VMSSASIN

contains the value of the SYSIN= system option and provides you with the name of the SAS job that is currently being run. When the SAS System is run in interactive mode, the value of VMSSASIN is SYS$INPUT.

The following is an example using this macro variable:

```
data test;
   infile '[school]roster.dat';
```

```
   input name $ age grade $;
run;

proc print data=test;
   title "Output generated by &vmssasin program.";
run;
```

Alternatively, you could put the value of the VMSSASIN macro variable into a variable in your data set:

```
data test;
   infile '[school]roster.dat';
   input name $ age grade $;
   job=symget("vmssasin");
run;
```

# Macro Statements

The following macro statements have operating dependencies that are specific to the OpenVMS environment:

%KEYDEF
is analogous to the KEYDEF command in the SAS windowing environment. It enables you to define function keys. The syntax of this statement is

KEYDEF *key-name* | *'key-name'* <*'definition'*>;

If you omit the definition, SAS prints a message in the log showing the current definition of the key; otherwise, the key definition is changed to whatever you specified. The value of *key-name* varies from terminal to terminal. You can define any key that is listed in the KEYS window. Any keyname that is hyphenated or contains a space, such as "CTRL-G" and "Gold 9", must be enclosed in double quotation marks.

%SYSEXEC
issues DCL commands immediately and places the operating environment return code in the automatic variable SYSRC. The syntax of the %SYSEXEC statement is

%SYSEXEC <*DCL-command*>;

where *DCL-command* can be any OpenVMS operating environment command or any sequence of macro operations that generates an operating environment command.

The %SYSEXEC statement is similar to the X statement, which is described in "Issuing DCL Commands during a SAS Session" on page 36. You can use the %SYSEXEC statement either inside a macro or in open code.

Omitting *DCL-command* puts you in an interactive OpenVMS subprocess and sets the value of the SYSRC automatic variable to 0. To return to your SAS session, type **LOGOFF** at the subprocess prompt.

The following is an example of %SYSEXEC:

```
%sysexec show time;
```

The output looks something like this:

```
12-JAN-1998 16:02:52
```

# Macro Functions

The following macro function has details that are specific to the OpenVMS operating environment:

%SYSGET
returns the character-string value of the OpenVMS symbol that you specified as the argument. The syntax of this function is

%SYSGET(*OpenVMS-symbol-name*);

You can use %SYSGET to translate either local or global OpenVMS symbols. If the symbol that you specify does not exist, SAS prints a warning message in the log.

## Example: Using the %SYSGET Function

The following example writes square brackets ( **[]**) to the SAS log.

1 Issue the following command to define a global symbol, HERE, to be **[]**:

```
$ HERE == "[]"
```

2 Invoke the SAS System, using the invocation command that is used at your site (usually **$ SAS**).

3 In your SAS session, assign the value of the %SYSGET function to the macro variable VAR1, using the symbol HERE as the argument.

```
%let var1=%sysget(here);
%put &var1;
```

# Autocall Libraries

An autocall library contains files or members that define SAS macros. The autocall facility enables you to invoke a macro without having previously defined that macro in the same SAS program. In order to use the autocall facility, the SAS system option MAUTOSOURCE must be in effect. (For information about the MAUTOSOURCE system option, see *SAS Language Reference: Dictionary*.)

SAS Institute supplies some autocall macros. When the SAS System is installed, a SASAUTOS logical name is defined. This OpenVMS logical name refers to the location of the default macros that are supplied by SAS Institute. Whether this logical name is placed in the system-level logical name table or in the process-level logical name table is site-dependent.

You can also define your own autocall macros in a user autocall library.

## Creating an Autocall Macro

To create an autocall macro, perform the following tasks:

1 Create either an OpenVMS directory or an OpenVMS text library to function as an autocall library, or use an existing autocall library.

2 In the autocall library, create a file (filetype .SAS) or member (filetype .TLB) that contains the source statements for the macro. The filename or member name must be the same as the name of the macro. For example, if a file named

PRTDATA.SAS is stored in an OpenVMS directory, then the file must define a macro named PRTDATA. Similarly, if the text library MYLIB.TLB contains the member DATACONT, then that member must define a macro named DATACONT.

## Specifying a User Autocall Library

Use the SAS system option SASAUTOS to specify the location of one or more user autocall libraries. (For more information about this option, see "SASAUTOS=" on page 434.) You can specify autocall libraries either when you invoke SAS or during a SAS session, as follows:

- When you invoke SAS:

  single autocall library:

  ```
  SAS/MAUTOSOURCE/SASAUTOS='[mydir]'
     [dir]program
  ```

  concatenated autocall library:

  ```
  SAS/MAUTOSOURCE/SASAUTOS=('[mydir1]',
     '[mydir2]',sasautos) [dir]program
  ```

  *Note:* Invocation options are separated by slashes following the SAS command. △

- During a SAS session (using an OPTIONS statement in your program):

  single autocall library:

  ```
  options mautosource sasautos='[mydir]';
  ```

  concatenated autocall library:

  ```
  options mautosource sasautos=('[mydir1]',
     '[mydir2]',sasautos);
  ```

# Stored Compiled Macro Facility

The stored compiled macro facility gives you access to permanent SAS catalogs that contain compiled macros. In order for SAS to use stored compiled macros, the SAS system option MSTORED must be in effect. In addition, you use the SAS system option SASMSTORE= to specify the libref of a SAS data library that contains a catalog of stored compiled SAS macros. For more information about these options, see *SAS Language Reference: Concepts*.

Using stored compiled macros offers the following advantages over other methods of making macros available to your session:

- SAS does not have to compile a macro definition when a macro call is made.
- Session-compiled macros and the autocall facility are also available in the same session.

Because you cannot re-create the source statements from a compiled macro, you must save the original macro source statements.

*Note:* Catalogs of stored compiled macros cannot be concatenated. △

If you do not want to use the stored compiled macro facility, you can make macros accessible to your SAS session or job by doing the following:

- placing all macro definitions in the program before calling them
- using a %INCLUDE statement to bring macro definitions into the program from external files
- using the autocall facility to search predefined source libraries for macro definitions.

Your most efficient choice may be to use the stored compiled macro facility.

## Accessing Stored Compiled Macros

The following example illustrates how to create a stored compiled macro in one session and then use the macro in a later session:

```
/*  Create a Stored Compiled Macro  */
/*      in One Session              */
libname mylib '[dir]';
options mstored sasmstore=mylib;

%macro myfiles / store;
   filename file1 '[dir]first.dat';
   filename file2 '[dir]second.dat';
%mend;

/*  Use a Stored Compiled Macro     */
/*     in a Later Session           */
libname mylib '[dir]';
options mstored sasmstore=mylib;
%myfiles
data _null_;
   infile file1;
      *statements that read input file FILE1;
   file file2;
      *statements that write to output file FILE2;
run;
```

## Controlling Memory Availability for Storing Macro Variables

Two system options control the maximum amount of memory available for storage of macro variables:

MVARSIZE=*n*
: specifies the maximum number of bytes for any macro variable stored in memory (0 <= *n* <= 32768). The default setting for this option in the OpenVMS operating environment is 8192.

MSYMTABMAX=*n*
: specifies the maximum amount of memory available to all symbol tables (global and local combined). The value of *n* can be expressed as an integer or MAX (the largest integer your operating environment can represent). The default setting for this option in the OpenVMS operating environment is 51200.

You can specify these system options in the following places:

- at SAS invocation
- in the configuration file

- during execution with an OPTIONS statement or in the System Options window

# Other Host-Specific Aspects of the Macro Facility

## Collating Sequence for Macro Character Evaluation

Under OpenVMS, the macro facility uses ASCII collating sequences for %EVAL, %sysevalf, and for implicit evaluation of macro characters.

## SAS System Options Used by the Macro Facility

Table 19.1 on page 467 lists the SAS system options that are used by the macro facility and that have host-specific characteristics. The table also tells you where to look for more information about these system options.

**Table 19.1** SAS System Options Used by the Macro Facility That Have Host-Specific Aspects

| System Option | Description | See |
|---|---|---|
| MSYMTABMAX= | Specifies the maximum amount of memory available to all symbol tables (global and local combined). Under OpenVMS, the default value for this option is 51,200 bytes. | System option: "MSYMTABMAX=" on page 424 |
| MVARSIZE= | Specifies the maximum number of bytes for any macro variable stored in memory (0 <= *n* <= 32767). Under OpenVMS, the default setting for this option is 8,192. | System option: "MVARSIZE=" on page 425 |
| SASAUTOS= | Specifies the autocall library. | System option: "SASAUTOS=" on page 434 and "Specifying a User Autocall Library" on page 465 |

# See Also

- *SAS Macro Language: Reference*
- *SAS Macro Facility Tips and Techniques*
- the SAS online Help system.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OpenVMS Environment, Version 8,* Cary, NC: SAS Institute Inc., 1999. 518 pp.

**SAS® Companion for the OpenVMS Environment, Version 8**



1–58025–526–4



1st printing, October 1999



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

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