CHAPTER

14 Macro Facility

About the Macro Facility

Most features of the SAS macro facility are portable. This chapter discusses only those components of the macro facility that depend on the UNIX environment. For more information, refer to

- *SAS Macro Language: Reference*
- *SAS Macro Facility Tips and Techniques*
- the online help for the macro facility.

Automatic Macro Variables

The following automatic macro variables are portable, but their values are determined by the operating environment:

SYSCC
: contains the current SAS condition code that SAS will return to UNIX when the SAS System exits. Upon exit, SAS translates this condition code to a return code that has a meaningful value for the operating environment. Under UNIX, SYSCC may contain the following condition codes:

0	Normal completion
1	SAS issued warning(s)
2	SAS issued error(s)
3	ABORT;
4	ABORT RETURN *n*;
5	ABORT ABEND *n*;

6	Internal error

SYSDEVIC
contains the name of the current graphics device. The current graphics device is determined by the DEVICE system option. Contact your SAS support representative to determine which graphics devices are available at your site. (See "DEVICE" on page 263 and *SAS Language Reference: Dictionary* for information about the DEVICE system option.

SYSENV
contains the value **FORE** when the TERMINAL system option is in effect; **BACK**, otherwise.

SYSJOBID
lists the PID of the process that is executing the SAS System, for example, 00024.

SYSMAXLONG
returns the maximum long integer value allowed under UNIX, which is 2,147,483,647. On 64-bit systems, the maximum is 9,007,199,254,740,992.

SYSRC
holds the decimal value of the exit status code that was returned by the last UNIX command that was executed from your SAS session. Output 14.1 on page 214 shows an interactive line mode SAS session showing two sample SYSRC values.

Output 14.1 Sample SYSRC Values

```
1? x 'data';
/bin/ksh: data: not found
2? %put UNIX exit status code is &sysrc;
UNIX exit status code is 256
3? x 'date';
Tue Mar 23 09:41:27 CST 1993
4? %put UNIX exit status code is now &sysrc;
UNIX exit status code is now 0
```

SYSSCP
returns the abbreviation for your environment, such as **HP 800**, **SUN 4**, or **RS6000**.

SYSSCPL
returns the name of the specific UNIX environment that you are using, such as **HP-UX**, **SunOS**, or **AIX**.

Macro Statements

The arguments that can be entered with the following statements depend on the operating environment:

%KEYDEF
: is analogous to the KEYDEF command. It enables you to define function keys. The %KEYDEF statement has the following syntax:

%KEYDEF *keyname* | '*keyname*' <'*definition*'>;

If the definition is omitted, a message is printed to the log showing the current definition of the key; otherwise, the key's definition is changed to whatever you specify.

Key names vary from terminal to terminal. You can define any key listed in the KEYS window, provided that it is not reserved by UNIX. You must enclose in quotes any key name that is hyphenated or contains a space, such as CTRL A or SHFT F1. For example, to assign the RECALL command to the CTRL A key sequence, submit the following statement:

```
%keydef 'ctrl A' 'recall';
```

%SYSEXEC
: executes UNIX commands. It is similar to the X statement described in "Executing Operating System Commands from Your SAS Session" on page 12. The %SYSEXEC statement enables you to execute operating environment commands immediately and, if necessary, determine whether they executed successfully by examining the value of the automatic macro variable SYSRC. You can use the %SYSEXEC statement inside a macro or in open code. The form of the %SYSEXEC statement is as follows, where *command* can be any UNIX command:

%SYSEXEC <*command*>;

For example, the following code writes the status of the default printer to your UNIX shell:

```
%sysexec lpstat;
```

Entering %SYSEXEC without a UNIX command starts a new shell, except under the X Interface to the SAS System. See "Executing Operating System Commands from Your SAS Session" on page 12 for details.

Macro Functions

The following functions have operating environment dependencies:

%SCAN
: searches for a word that is specified by its position in a string. On ASCII systems, the default delimiters are

```
blank . < ( + & ! $ * ) ; ^ - / , % |
```

%SYSGET
: returns the character string that is the value of the environment variable passed as the argument. Both UNIX and SAS environment variables can be translated using the %SYSGET function. A warning message is printed if the global variable does not exist. The form of the %SYSGET function is

%SYSGET(*environment-variable*);

For example, the following code writes the value of the HOME environment variable to the SAS log:

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

SAS System Options Used by the Macro Facility

The following system options have operating environment dependencies:

MSYMTABMAX
: specifies the maximum amount of memory available to all symbol tables (global and local combined). Under UNIX, the default value for this option is 4M. See *SAS Language Reference: Dictionary* for more information.

MVARSIZE
: specifies the maximum number of bytes for any macro variable stored in memory. Under UNIX, the default value for this option is 32K. See *SAS Language Reference: Dictionary* for more information.

SASAUTOS
: specifies the AUTOCALL library. See “The SASAUTOS System Option” on page 216 and “SASAUTOS” on page 288 for more information.

Using Autocall Libraries

An autocall library contains files that define SAS macros. The following sections discuss aspects of autocall libraries that are dependent on the operating environment. For more information, see *SAS Macro Language: Reference*.

There are two types of autocall macros, those provided by SAS Institute and those you define yourself. To use the autocall facility, you must have the system option MAUTOSOURCE set.

When the SAS System is installed, the SASAUTOS system option is defined in the configuration file to refer to the location of the default macros supplied by the Institute. The products licensed at your site determine the autocall macros you have available. You can also define your own autocall macros and store them in one or more directories.

If you store autocall macros in a UNIX directory, the file extension must be **`.sas`**. Each macro file in the directory must contain a macro definition with a macro name that matches the filename. For example, a file named **`prtdata.sas`** should define a macro named PRTDATA.

The SASAUTOS System Option

To use your own autocall macros in your SAS program, specify their directories with the SASAUTOS system option. See “SASAUTOS” on page 288 for a complete description of the SASAUTOS system option.

You can set the SASAUTOS system option when you start the SAS System, or you can use it in an OPTIONS statement during your SAS session. However, autocall libraries specified with the OPTIONS statement override any previous specification.

If you use the CONFIG system option to specify a configuration file, add your autocall library to the library concatenation supplied by SAS Institute. If you use the default configuration files (**config.sasv8**) simply specify your autocall library there.

Autocall libraries are searched in the order in which you specify them.

Example

This example shows how to set up and test a macro in an autocall library.

Output 14.2 on page 217 shows two UNIX (**cat**) commands to display the contents of two files and a SAS command to run the AUTOCALL.SAS program.

Output 14.2 AUTOCALL Library Example

```
$ cat maclib/testauto.sas
%macro testauto;
x echo 'Autocall library is working.';
%mend testauto;
$ cat source/autocall.sas
filename sysautos ('!SASROOT/sasautos' '$HOME/test/sasautos');
options mautosource sasautos=(sysautos '$HOME/macros/maclib');
%testauto
$ sas source/autocall.sas
Autocall library is working.
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for UNIX Environments, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for UNIX Environments, Version 8

ISBN 1–58025–502–7

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

1st printing, October 1999