CHAPTER 5

AF Window Commands

Overview

The following commands are available in the AF window in which SAS/AF applications execute. The behavior of the window depends on which type of catalog entry is being displayed. Some commands are not available for all catalog entry types. Exceptions are noted in the command descriptions.

Note: For FRAME and PROGRAM entries, application developers can provide additional commands that are specific to the entry.
△

Window Management	*=menu-option-list*	MUSIC
	AFSYS	PREVIEW
	CANCEL	QCAN
	CATNAME	QEND
	CLEAR	QSTACK
	DEBUG	RECALL
	DUP	RETURN|=
	END	SAVE
	FIND	SOUND
	HELP	TYPE
	ID	WREGION
	KEYFIELD	WSIZE
	LOCATE	
Scrolling	BACKWARD	LEFT
	BOTTOM	RIGHT
	FORWARD	TOP
	HSCROLL	VSCROLL
Printing	FONT	PRTFILE
	FORMNAME	SPRINT

In addition to the commands shown in this list, all of the SAS windowing environment global commands are also available in the AF window. Refer to the online Help for base SAS software for descriptions of the global commands.

Note: Application developers can use SCL programs in FRAME and PROGRAM entries to block the execution of some or all of these commands, or to provide different processing of the commands. △

Window Management Commands

=*menu-options*

executes the entry called by the specified options for the first MENU entry in the current execution stack. To specify submenu options in the *menu-options* value, separate the options with periods.

For example, if the application window currently displays a PROGRAM entry that was opened from a MENU entry, the following command attempts to open the entry that is selected by the option value 5 in the original MENU entry:

```
=5
```

AFSYS *action*

enables you to review and manage resident entries and to control the automatic termination of FRAME objects that still exist when a FRAME entry ends. This command is useful for testing applications.

The AFSYS command uses the following *action* arguments:

AUTOTERM ON | OFF

controls whether the _term method of a FRAME entry object is automatically executed if the object still exists when a FRAME entry ends. By default, any open objects are automatically terminated (AUTOTERM ON). Use AUTOTERM OFF if you want to prevent the _term method of open FRAME objects from being executed when the FRAME entry ends.

AUTOTERM VERBOSE | NOVERBOSE>
controls whether a list of open objects is sent to the Log window. By default, no list is generated (AUTOTERM NOVERBOSE). Use AUTOTERM VERBOSE to send the object list for each object that still exists when the FRAME entry ends.

SHOW ACTIVE
sends a list of all entries that are active for the current application to the Log window.

SHOW INACTIVE
sends a list of all entries that are resident but inactive for the current application to the Log window.

RESIDENT ON | OFF
controls whether entries that have the RESIDENT attribute are retained in memory when they are inactive. The default is to retain entries (RESIDENT ON). Specify OFF to prevent inactive entries from being retained in memory even if they have the RESIDENT attribute.

PURGE *n* | ALL>
removes inactive entries from memory. Specify ALL to remove all inactive entries. Specify an entry number to remove a particular inactive entry. Use the AFSYS SHOW INACTIVE command to view the entry numbers for inactive entries.

To specify a combination of actions, use separate AFSYS commands, as shown in the following example:

```
afsys autoterm verbose; afsys autoterm on
```

CANCEL
closes the current entry and returns control to the calling entry, or to the parent entry if one was specified in the current entry's general attributes.

For FRAME and PROGRAM entries, the CANCEL command performs the following tasks before closing the entry:

□ sets the SCL _STATUS_ variable to **C**

□ runs the TERM section of the entry's SCL program (and, for FRAME entries, the _term method for the frame and all components).

Any pending SAS statements in the PREVIEW buffer are not submitted to the SAS System for processing.

CATNAME <CLEAR | LIST> *<libref.>catalog-reference* (*<libref.>catalog-1* ... *<libref.>catalog-n*)
defines a catalog reference (catref) that provides a logical combination of the specified catalogs. You can use any valid SAS name for *catalog-reference*, but if you use the name of an existing catalog you will not be able to access the contents of that catalog until the catref is cleared. The libref that you specify with the catref must already exist. Enclose the list of catalogs in parentheses, and use blanks to separate the catalog names in the list.

When a program in the SAS/AF application contains a reference to an entry in the catref catalog, the AF task searches for the entry in the specified list of catalogs, starting with *catalog-1* and ending with *catalog-n*.

For example, suppose that you issue the following command:

```
catname mylib.all (mylib.apps1 mylib.apps2 master.apps)
```

A reference in a program to MYLIB.ALL.TEST.SCL causes the AF task to search for MYLIB.APPS1.TEST.SCL, then for MYLIB.APPS2.TEST.SCL, and finally for MASTER.APPS.TEST.SCL.

Catref assignments remain in effect until they are cleared. Use the CLEAR option with the CATNAME command to clear a specified catref. Use the LIST option with the CATNAME command to list the catalogs in the catref.

CLEAR

clears values from all unprotected data entry areas of the application window.

Note: The CLEAR command is only supported for PROGRAM entries. △

DEBUG <ON | OFF>

turns the SAS Component Language source-level debugger on or off. If you use the DEBUG command without an argument, the debugger is turned on. You cannot turn the debugger off while a debugger command is active.

Note: In order to use the debugger, the SCL code in the entry must have been compiled with the DEBUG compile option turned on. △

The SCL debugger is a tool for identifying and correcting problems in SAS Component Language programs. Refer to *SAS Component Language: Reference* for information on using the SCL debugger.

DUP

copies the value stored by the SELECT command to the current field. The cursor must be positioned on the desired field when you issue the command, so the DUP command is easier to use if you assign it to a function key.

Note: The DUP command is only supported for PROGRAM entries. △

END

closes the current entry and returns control to the calling entry, or to the parent entry if one was specified in the current entry's general attributes.

For FRAME and PROGRAM entries, the END command performs the following actions before closing the entry:

- □ sets the SCL _STATUS_ variable to **`E`**
- □ verifies that modified fields contain valid values
- □ executes the MAIN section of the SCL program if any fields have been modified and contain valid values
- □ verifies that all required fields contain values
- □ executes the TERM section of the SCL program (and, for FRAME entries, the _term method for the frame and all components)
- □ submits the contents of the PREVIEW buffer, including any submit blocks in the TERM section.

FIND *search-string* NEXT | FIRST | LAST | PREV | ALL> <PREFIX | SUFFIX | WORD> <CASE | ICASE>

scrolls the line that contains the specified value to the top of the window for MENU and HELP entries.

For PROGRAM entries that contain extended tables, the FIND command finds an occurrence of the specified value in the field that is identified with the

KEYFIELD command. If the string contains embedded blanks, it must be enclosed in quotes.

You can modify the behavior of the FIND command by adding any one of the following options:

ALL	reports the total number of occurrences of the string in the extended table on the window's message line and moves the cursor to the first occurrence.
FIRST	moves the cursor to the first occurrence of the string in the extended table .
LAST	moves the cursor to the last occurrence of the string in the extended table.
NEXT	moves the cursor to the next occurrence of the string in the extended table.
PREV	moves the cursor to the previous occurrence of the string in the extended table.

The default behavior is NEXT.

By default, the FIND command locates any occurrence of the specified string, even where the string is embedded in other strings. You can use any one of the following options to change the command's behavior:

PREFIX	causes the search string to match the text string only when the text string occurs at the beginning of a word.
SUFFIX	causes the search string to match the text string only when the text string occurs at the end of a word.
WORD	causes the search string to match the text string only when the text string is a distinct word.

By default, the FIND command is case-sensitive. You can use the following options to change the command's behavior:

CASE	causes the search string to match the text string only if the case is the same.
ICASE	causes the search string to match the text string regardless of case.

After you issue a FIND command, you can use the RFIND command to repeat the search for the next occurrence of the string, or use the BFIND command to repeat the search for the previous occurrence.

Note: The FIND command is only supported for HELP, MENU, and PROGRAM entries. △

HELP

opens the entry specified in the current entry's Help attribute.

ID <ON | OFF>

displays the four-level name of the current entry on the window's message line for FRAME, HELP, MENU, or PROGRAM entries. For FRAME entries, the four-level name of the associated SCL entry is also displayed.

For CBT entries, the ID command shows the current frame number and frame name in the window title. Once turned on, the ID information is displayed until you turn it off. If you use the ID command without the ON or OFF option, it acts as a toggle.

Note: The ON and OFF options are only supported for CBT entries. △

KEYFIELD

identifies the field that is searched by subsequent FIND and LOCATE commands. The cursor must be positioned on the desired field when you issue the command, so the KEYFIELD command is easier to use if you assign it to a function key.

Note: The KEYFIELD command is only supported for PROGRAM entries. △

LOCATE <:> *search-string*

finds an occurrence of the specified value in the extended table field that is identified with the KEYFIELD command. If the string contains embedded blanks, special characters, or lowercase letters, then it must be enclosed in quotes.

The LOCATE command always begins matching from the first character of the field value. By default, it searches only for an exact match of the entire field value. To match only the first part of a field value, add a colon (:) before the search string.

After you issue a LOCATE command, you can use the RLOCATE command to repeat the search for the next occurrence of the string, or use the BLOCATE command to repeat the search for the previous occurrence.

Note: The LOCATE command is only supported for PROGRAM entries that display extended tables. △

MUSIC

See the SOUND command.

PREVIEW

opens the PREVIEW window, in which you can view and edit any SAS statements that are waiting to be submitted for processing when the application ends. The SAS statements are generated by SUBMIT blocks in SAS Component Language programs. Refer to *SAS Component Language: Reference* for more information on SUBMIT blocks.

QCAN

closes all open entries for the current application and returns control to the SAS System.

For FRAME and PROGRAM entries, the QCAN command performs the following steps before closing the entry:

□ sets the SCL _STATUS_ variable to **C**

□ runs the TERM section of the entry's SCL program (and, for FRAME entries, the _term method for the frame and all components).

Any pending SAS statements in the PREVIEW buffer are not submitted to the SAS System for processing.

Note: The difference between the QCAN command and the CANCEL command is that the QCAN command always returns control to the SAS System, not to the calling entry or the parent entry. △

QEND

closes all open entries for the current application and returns control to the SAS System.

For FRAME and PROGRAM entries, the QEND command performs the following actions before returning control to the SAS System:

□ sets the SCL _STATUS_ variable to **E**

□ executes the MAIN section of the SCL program if a field has been modified

□ executes the TERM section of the SCL program (and, for FRAME entries, the _term method for the frame and all components)

□ submits the contents of the PREVIEW buffer, including any submit blocks in the TERM section

Note: The difference between the QEND command and the END command is that the QEND command always returns control to the SAS System, not to the calling entry or the parent entry. △

QSTACK

prints a report of the application's execution stack in the Log window. The stack contains the active entry that is currently running and all the inactive entries that have been opened since you started executing the application. The listing shows two parts of the execution stack:

N-Stack	lists entries that can be viewed with the NEXT command.
S-Stack	lists all the entries in the execution stack. This stack lists interim MENU entries that have been called but are inactive and cannot be displayed with the NEXT command.

RECALL

recalls any field values that have been saved with the SAVE command (or with the SAVESCREEN function in an SCL program or with the AUTOSAVE=YES option in the AF command).

Note: The RECALL command is only supported for PROGRAM entries. △

RETURN
=

opens the entry at which the application started executing.

Note: When the RETURN command is issued in a window that was opened by a CALL GOTO routine in SCL, the result depends on the action option specified in the routine. △

SAVE

saves the current values of a PROGRAM entry's fields. You can recall the saved values later by using the RECALL command. The values are stored in an entry named *program-name*.AFPGM in your SASUSER.PROFILE catalog, where *program-name* is the name of the PROGRAM entry.

For CBT entries, the SAVE command records the current entry name and frame number, then ends the application and the SAS session. You can issue the AF command in a later SAS session to resume the CBT entry at the frame from which you issued the SAVE command.

Note: The SAVE command is only supported for CBT and PROGRAM entries. △

SOUND <ON | OFF>
MUSIC <ON | OFF>

controls whether sounds are produced for SOUND= or MUSIC= options in frames of CBT entries (provided your display device is capable of producing sounds). Use SOUND OFF if you do not want the hear the sounds that are specified in the CBT frame. If you issue a SOUND command without an ON or OFF option, it acts as a toggle, turning sound off if it was on or on if it was off.

Note: The SOUND command is only supported for CBT entries. △

TYPE <*entry-type*>

specifies the entry type that is assumed when the type is not explicitly specified in a command that must ordinarily be followed by an entry name in the form *entry-name.entry-type*. Use the TYPE command without arguments to display the current default type on the window's message line.

WREGION <*start-row start-column rows columns*>
WREGION <<TOP | BOTTOM | LEFT | RIGHT> <DEVICE>>
WREGION CLEAR

specifies the position and size of the next window of a multi-window session.

The *start-row* and *start-column* values specify the position of the upper left corner of the window, and the *rows* and *columns* values specify the window size.

Note: You can use the WSIZE command to determine the current window specifications before you issue a WREGION command. △

Alternatively, you can specify TOP, BOTTOM, LEFT, or RIGHT to cause the next window to occupy the corresponding half of the current window. For example, WREGION TOP specifies that the next window occupies the top half of the current window. Add the DEVICE option to indicate that the specification is relative to the device display rather than to the current window. For example, WREGION TOP DEVICE specifies that the next window occupies the top half of the device's display. This option is not supported in some display environments.

The CLEAR option clears any previous WREGION setting and returns the window to the size specified in the entry's general attributes. If the size is not specified in the general attributes, the window size returns to the default size.

WSIZE

displays the size and position specifications of the current window on the window's message line.

Scrolling Commands

BACKWARD <*n* | HALF | PAGE | MAX>

scrolls toward the top of the HELP, MENU, or PROGRAM entry's display text if the text contains more lines than the current window size. If you omit the scroll increment, the window is scrolled by a default increment that can be set with the VSCROLL command.

For CBT entries, the BACKWARD command displays the previous frame in the current sequence.

Note: The BACKWARD command is not supported for FRAME entries. △

BOTTOM

scrolls the window contents so that the last line of the display text appears in the window.

Note: The BOTTOM command is only supported for HELP, MENU, and PROGRAM entries. △

FORWARD <*n* | HALF | PAGE | MAX>

scrolls toward the bottom of the HELP, MENU, or PROGRAM entry's display text if the text contains more lines than the current window size. If you omit the scroll increment, the window is scrolled by a default increment that can be set with the VSCROLL command.

For CBT entries, the FORWARD command displays the next frame in the current sequence.

Note: The FORWARD command is not supported for FRAME entries. △

HSCROLL <*n* | HALF | PAGE>

sets the scroll amount for LEFT or RIGHT commands that do not specify a scroll increment. The default horizontal scroll increment is HALF.

Note: The HSCROLL command is only supported for HELP, MENU, and PROGRAM entries. △

LEFT <*n* | HALF | PAGE>
scrolls the window contents to the left by the specified increment. If you omit the scroll increment, the window is scrolled by a default increment that can be set with the HSCROLL command.

Note: The LEFT command is not supported for FRAME entries. △

RIGHT <*n* | HALF | PAGE>
scrolls the window contents to the right by the specified increment. If you omit the scroll increment, the window is scrolled by a default increment that can be set with the HSCROLL command.

Note: The RIGHT command is not supported for FRAME entries. △

TOP
scrolls the window contents so that the first line of display text appears at the top of the window.

Note: The TOP command is only supported for HELP, MENU, and PROGRAM entries. △

VSCROLL <*n* | HALF | PAGE | MAX>
sets the scroll amount for BACKWARD or FORWARD commands that do not specify a scroll increment. The default vertical scroll increment is PAGE.

Note: The VSCROLL command is only supported for HELP, MENU, and PROGRAM entries. △

Printing Commands

FONT
opens the FONT window and displays the font-control information from the current form. The listing shows which color and highlighting attributes the form interprets as signals to change printing characteristics. The FONT window is for browsing only; to change the font information you must open the FORM window and edit the FORM entry. See the description of the FORM window in base SAS documentation for more information.

Note: The FONT window is not opened if no font information is defined in the current form. Instead, a message is displayed, indicating that the form contains no font information. △

FORMNAME <*form-name*>
FORMNAME CLEAR
specifies the default FORM entry, which contains instructions for printing images that are captured with the SPRINT command. The SAS System's default form is DEFAULT.FORM. You can use the FORMNAME command to specify a different default. See base SAS documentation for more information about forms.

When a form is required, the SAS System first looks for the specified FORM entry in the current catalog. If the form is not there, the system next looks in your personal PROFILE catalog (SASUSER.PROFILE, or WORK.PROFILE if the SASUSER library is not defined) and then, finally, in the SASHELP.FSP catalog.

Use the FORMNAME CLEAR command to resume using DEFAULT.FORM as the default FORM entry name.

Use the FORMNAME command with no arguments to display the current default form on the window's message line.

PRTFILE <*fileref* | '*actual-filename*' <APPEND | REPLACE>>
PRTFILE CLEAR

specifies a file to which output from the SPRINT command is sent. By default, output is sent to the printer destination that is specified in the current form. You can use the PRTFILE command to route the output to a file instead.

To identify the target file, you can use either a previously assigned fileref or the actual filename. If you specify a filename, it must be enclosed in quotes.

With the filename or fileref, you can also specify either the APPEND or REPLACE option to determine how output is handled when the file already exists. The default is REPLACE, which causes output that is sent to an existing file to overwrite the current contents of the file. To append the new output to any existing contents, use the APPEND option instead.

Use the PRTFILE CLEAR command to cancel the previous PRTFILE command and to route output to the printer again.

Use the PRTFILE command with no arguments to display the name of the current print file on the window's message line.

SPRINT <FILE=*fileref* | '*actual-filename*'> <FORM=*form-name*> <NOBORDER>
SPRINT FREE

captures the contents of the current window (except for the command and message lines). Unless you use the NOBORDER option, the window borders are also included in the capture.

Output characteristics are determined by an associated FORM entry. You can use a FORMNAME command before issuing a SPRINT command to change the default FORM entry for all captures. Use the FORM= option with the SPRINT command to select a form other than the default for an individual capture.

By default, output is sent to the printer destination that is specified in the associated form. Use the PRTFILE command before issuing a SPRINT command if you want to send all output to a file instead of to the printer. To route an individual capture to a file instead of to the printer, or to route an individual capture to a file other than the one specified in the PRTFILE command, use the FILE= option with the SPRINT command. Changing output destinations within an application automatically frees the previous print file or print queue.

Note: Once you have sent SPRINT output to a file, any additional output that you send to that file must use the same FORM entry. △

The print queue or print file that the SPRINT command uses is freed when you end the application from which you captured contents. To free the print queue or print file before ending the application, use the SPRINT FREE command.

You can use the SPRINT command to capture information from several windows in a single print file or print queue. In this case, the print file or print queue is not freed until you end the application that sent output to the file or queue.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/AF® Software Procedure Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/AF® Software Procedure Guide, Version 8

ISBN 1–58025–516–7

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

1st printing, October 1999

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