uni-SPF®

Reference Manual

525 Capital Drive
Lake Zurich, IL 60047
Telephone: (800) 228-0255
Fax: (847) 540-5629
Email: tech@wrkgrp.com
http://www.wrkgrp.com/

Preface

This manual describes the features of uni-SPF®, a panel driven productivity tool for the Unix environment. uni-SPF is patterned after IBM's ISPF Program Product. This manual corresponds to release 1.33 of uni-SPF.

This manual refers to uni-XEDIT®. uni-XEDIT is a full-screen editor marketed by The Workstation Group that is patterned after IBM's VM/CMS System Product Editor.

Document Number: SPFV1-3

Table of Contents

Chapter 1: Introduction

uni-SPF from The Workstation Group provides familiar tools and services for individuals migrating from IBM's MVS or VM operating system to a distributed processing environment based on Unix workstations. It pro-vides many functions similar to IBM's ISPF Program Product, including

- Panel-driven user interface
- Full function editor
- File browse feature
- Utilities for file and directory management
- Menus for foreground and background processes
- Access to the Unix shell
- Dialog test

Those familiar with ISPF will be immediately productive in the uni-SPF environment. Others will find the hierarchical structure and menu-based style of the panels intuitive and easy to use without exhaustive study of the documentation.

The uni-SPF reference manual fully describes all features of uni-SPF. It is organized according to the hierarchical structure of the panel-driven interface.

Documentation Conventions

The following conventions are used throughout this document to facilitate command syntax descriptions.

Uppercase

Uppercase letters indicate commands, keywords, or portions thereof that must be typed exactly as shown. You may type the command or keyword in any case, regardless of the documentation convention used.

Lowercase

Lowercase letters indicate variable information that you supply. A single character (usually ***n***) represents a number that you specify. Other variable data is represented by a descriptive name such as ***string***, ***end-column***, or ***range***. Variable data is also italicized to facilitate references to it within descriptive text.

Abbreviations

Some commands and keywords may be abbreviated. When a single abbreviation is allowed, this is indicated by uppercase letters. For example, the keyword "label" of the RESET command may be entered as LABEL or LAB. It is shown in the syntax diagram as **LABel**.

When multiple abbreviations are allowed, all possible abbreviations are stacked in the diagram as for the BOUNDS command:

BOUNDS [*left-column right-column*]
BOUND
BNDS
BND

Only one form of the command need be entered.

Optional Operands

Many commands have optional keywords or operands. These are shown within brackets in the syntax diagram. The BOUNDS example above illustrates this. When an operand may have more than one value, the options are stacked within brackets as in the DOWN command:

DOWN **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

You actually type only one of the choices – for example, DOWN MAX. Note the use of uppercase letters to indicate the acceptable abbreviation of each optional operand.

Required Operands

Required operands are shown without brackets as in the DELETE command:

DELETE ALL [X] [*range*]
[NX]

When a required operand may have more than one value, the options are stacked in the same manner as for optional operands. As with optional operands, you type only one of the choices.

Repeating Operands

An ellipsis (. . .) in a syntax diagram indicates that an operand may be repeated one or more times. This is illustrated by the SORT command

SORT [*range*] [*sort-field-1* [. . . *sort-field-n*]]

where you may specify multiple sort-fields. Do not include the ellipsis when typing your command.

Delimiters

Command operands are normally delimited by a blank space. If any other delimiter is required or optional, it is mentioned specifically in the discussion of an indi-vidual command. Operands which include blank spaces (such as character strings used in FIND or CHANGE commands) should be enclosed in quotes to designate the beginning and end of the operand. This use of quotes is not shown in any syntax diagrams.

Hierarchical Structure

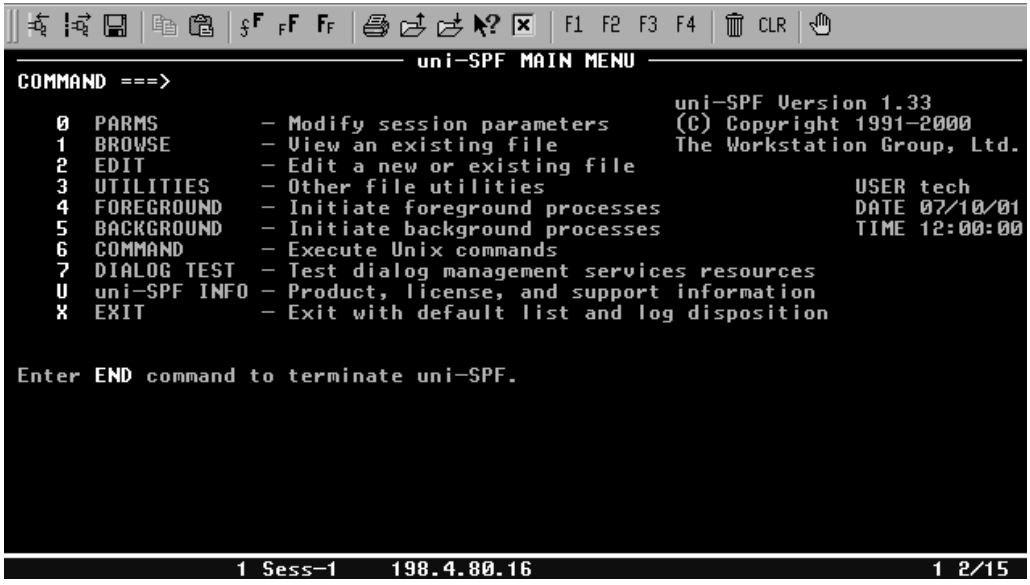
The uni-SPF facilities are organized into a hierarchical structure, with similar functions grouped together. A menu at each level allows you to choose the desired function by typing your selection on the OPTION or COMMAND line of the panel. Subsequent panels may contain menus, entry fields, or selection lists. In some cases, entry fields are automatically filled in by the system from data you have previously provided or from system defaults. You may change any of these entries by overtyping.

From any level in the hierarchy the END command (normally PF3) returns you to the previous level. Thus, you may navigate the system entirely through the use of menu selections and END.

The `=' command provides an alternative for proficient users to go directly from any point in the system to the specific service desired. For example, "=2" takes you immediately to the Edit-Entry Panel, or "=x" immediately ends uni-SPF, no matter where you are at the time you enter the directive.

The RETURN command (normally PF4) takes you immediately to the Main Menu panel from any point in the system.

The Main Menu panel is the top level of the hierarchy. It is a menu from which you select the facility that you wish to use.



The chapters of this reference manual correspond to the selections available on this panel, with the exception that Option U is described in ***Chapter 2, Interaction with uni-SPF***.

Each primary option may have one or more secondary options. To minimize the learning curve for those already familiar with ISPF, options that perform a function similar to ISPF have the same menu selection number that they have in ISPF. In some cases, this results in menus in which numbering does not begin with 1 or may not be sequential. The table on the following pages shows the complete hierarchy.

Option 0 - PARMS set uni-SPF session parameters

1 - TERMINAL specify selected display defaults

2 - LIST/LOG specify list and log defaults

3 - PF KEYS redefine program function keys

7 - PROFILE define or modify edit profiles

K - KEYBOARD MAPPINGS view, define, or modify keyboard mappings

Option 1 - BROWSE browse a file

A file selection panel appears if you do not specify a file name on the Browse-Entry Panel.

Option 2 - EDIT edit one or more files

A file selection panel appears if you do not specify a file name on the Edit-Entry Panel.

Option 3 - UTILITIES manipulate files and directories

1 - DIRECTORY operations on files in the directory specified

These include browse, edit, print, copy, delete, rename, compress/uncompress, and make directory. You can also change directories or modify file permissions. A file selection panel appears if you do not specify a file name for the operation chosen.

3 - MOVE/COPY copy or move files or directory structures

A file selection panel appears if you do not specify a file name for the operation chosen.

F - FILE LOCATOR
find files by name, pattern or other criteria; optionally, execute commands on files found

A file selection panel displays the results of the search.

J - JOB STATUS review status of submitted jobs

S - SEARCH FILES
find data in a directory or its subdirectories

T - TAPE ARCHIVE
operations on Unix tape archive (tar) files

Option 4 - FOREGROUND
execute foreground tasks

A secondary panel is available to initially define the applications or to modify an existing definition.

Option 5 - BACKGROUND
execute background tasks

A secondary panel is available to initially define the applications or to modify an existing definition.

Option 6 - COMMAND
execute a Unix command, program or shell script

Option 7 - DIALOG TEST
perform dialog testing

1 - TEST PANEL test panel definitions before installing

2 - TEST REXX DIALOG
test REXX dialog management programs before installing

C - TEST COMPILED LANGUAGE APPLICATION
build and test dialogs written in a compiled language

Option U - uni-SPF INFO
display product and license information

Option X - EXIT terminate the uni-SPF session

Chapter 2: Interaction with uni-SPF

Interaction with uni-SPF is accomplished through a series of panels. User response at each panel may be a simple menu selection or may involve specifying file or directory names or other requested information.

Proficient use of the system requires a thorough understanding of keyboard functions and, in some cases, knowledge of how to establish a keyboard mapping to insure that all the desired functions are available. It also requires an understanding of the various panels and of available uni-SPF commands. This chapter covers the details of all these topics.

In the area of keyboard functions, there may be some system dependencies among workstation types and terminal types. In such cases, this chapter discusses the general performance of the system and refers you to ***Appendix A: System Dependencies*** for the details related to your workstation and/or terminal.

Executing uni-SPF

Executing uni-SPF requires the setting of certain Unix environment variables. A shell script, ispf, is provided for this purpose. You must be sure that this shell script is located in a directory that is referenced in your PATH environment variable.

You may start uni-SPF in one of two ways:

`ispf` initiates uni-SPF and displays the Main Menu panel

`ispf` *`file1`* `[...` *`filen`*`]`
initiates uni-SPF and places you directly in the editor, with all of the files named in the edit ring. ***Chapter 5: Edit*** contains a discussion of the edit ring, which permits simultaneous editing of several files.

The ispf shell script sets the following environment variables:

SPFMSGPATH
directories where message files are stored

SPFPANELPATH
directories containing panel definition files

SPFREXXPATH
directories containing REXX dialog management programs

SPFTABLEPATH
directories where private tables are stored

SPFTABLEDIR
directory where new tables are stored

SPFHOME
HOME directory for file access and storage

SPFEDITPATH
directories containing editor macros

XEDITPATH
directories containing internal system files

uni-SPF also requires certain tables for storing profile and parameter information that is unique to each user. These tables, which are created in each user's SPFHOME directory, are

SPFPROF.tbl
: parameters that are common among panels, such as directory path and file name

ISPEPROF.tbl
: edit profile data

ISPKBND.tbl
: keyboard mappings

ISPFGNDP.tbl
: foreground private task definitions

ISPBGNDP.tbl
: background private task definitions

ISPFLOC.tbl
: prune sets for the File Locator Utility

ISPJOBS.tbl
: status of submitted jobs

ISPC*xxxx*.tbl
: tables used for building/testing compiled language applications

If you remove any of these files, the corresponding data will not be present when you next use uni-SPF.

Keyboard Functions

uni-SPF provides keyboard functions similar to those available on an IBM 3270-type terminal. These include

- Program function keys (PF keys)
- Cursor control keys
- Hardware edit keys
- Special key

The keyboards on most terminals used with Unix workstations do not have a one-to-one correspondence with these 3270-style keys. uni-SPF therefore has a generic key sequence defined for each of these keyboard functions. Keyboards with similar keys or with keys that can be mapped to the desired function may use such keys instead of (or in addition to) the generic key sequences.

The generic key sequences may include modifier keys such as Escape (ESC), Control (CTL) or Shift (SHIFT). When the key sequence includes ESC, press the Escape key, and then press the key specified. When the key sequence includes CTL or SHIFT, hold down the first key while you press the second key.

The sections that follow identify the function, the key name (such as PF1), and the generic key sequence. The section on "Keyboard Mapping" discusses how to use similar keys on your keyboard. Workstation- or terminal-specific issues are discussed in ***Appendix A: System Dependencies***.

Program Function Keys

Program function keys have been defined to perform specific uni-SPF commands. The initial uni-SPF PF key definitions for PF13-24 are identical to the definitions for PF1-12. Throughout this documentation, the key names PF1 through PF12 are used to refer to the default PF key definitions. Although twenty-four PF keys are often available on 3270-style terminals, there may not be sufficient keys available on workstation terminals to define all twenty-four PF keys.

PF key definitions do not change as you move among the uni-SPF options. As an example, when PF8 is defined as DOWN (the default) it always scrolls down, whether you are in the editor or a file selection panel.

The default PF key definitions and the generic key sequence that activates each one are shown below.

PFKey	Generic Sequence	Definition	Function
PF1	ESC 1	HELP	displays long messages
PF2	ESC 2	SPLIT	splits the screen at cursor location
PF3	ESC 3	END	terminates the current activity
PF4	ESC 4	RETURN	returns to the Main Menu panel
PF5	ESC 5	RFIND	repeats the last FIND command
PF6	ESC 6	RCHANGE	repeats the last CHANGE command
PF7	ESC 7	UP	scrolls the display up
PF8	ESC 8	DOWN	scrolls the display down
PF9	ESC 9	SWAP	moves between split screens
PF10	ESC 0	LEFT	scrolls the display left
PF11	ESC -	RIGHT	scrolls the display right
PF12	ESC =	RETRIEVE	retrieves the last command executed
PF13	ESC !	HELP	same as PF1
PF14	ESC @	SPLIT	same as PF2
PF15	ESC #	END	same as PF3
PF16	ESC $	RETURN	same as PF4
PF17	ESC %	RFIND	same as PF5
PF18	ESC ^	RCHANGE	same as PF6
PF19	ESC &	UP	same as PF7
PF20	ESC *	DOWN	same as PF8
PF21	ESC (	SWAP	same as PF9
PF22	ESC)	LEFT	same as PF10
PF23	ESC _	RIGHT	same as PF 11
PF24	ESC +	RETRIEVE	same as PF12

You may redefine PF keys to commands that you use more frequently. You may also expand your PF key usage by eliminating the duplication between PF1-12 and PF13-24. Use Option 0.3 to modify your PF key definitions.

Some of the commands assigned to PF keys have optional command operands. To use optional operands with the PF key, type the operand on the command line before pressing the PF key. This is identical to typing

```
COMMAND operand
```

on the command line and pressing Enter.

On many terminal/workstation combinations, the keys labelled “F1", ”F2", and so on are automatically recognized as PF keys. This depends on the mapping of that terminal’s keyboard that is provided as part of the operating system by the workstation manufacturer. For a full discussion of keyboard mapping, refer to the section entitled “Keyboard Mapping” in this chapter. ***Appendix A: System Dependencies*** contains more detailed information about keyboard mappings that may be needed for specific terminal/workstation combinations. The section “Keyind Maintenance Utility” in ***Chapter 3: Parameters*** guides you in using the keyboard test and mapping utility, uni-KEY, to perform necessary keyboard mappings. uni-KEY is accessed through Option 0.K.2.

Cursor Control Keys

The cursor control keys include the arrow keys, Tab, Back-Tab, and Home. Arrow keys move the cursor on the screen one character position in the direction indicated. When the cursor reaches the bottom of the screen, it wraps to the top. Similarly, when it reaches the top, it wraps to the bottom. When the cursor reaches the right edge of the screen, it wraps to the beginning of the next line. Similarly, when it reaches the left edge of the screen, it wraps to the right end of the previous line.

The tab key moves the cursor to the beginning of the next input field. Back-tab moves the cursor to the beginning of the previous input field.

The HOME key takes the cursor immediately to the beginning of the first input field on the screen. This is normally the COMMAND or OPTION line.

On most terminal/workstation combinations, the cursor control keys are automatically recognized by uni-SPF. In some cases, however, the keyboard mappings pro-vided with the operating system by the workstation manufacturer may be incomplete. If the cursor control keys on your keyboard do not control the cursor cor-rectly, refer to ***Appendix A: System Dependencies*** as well as the section "Keybind Maintenance Utility" in ***Chapter 3: Parameters*** for guidance on mapping these keys.

Hardware Edit Keys

Several keys on the 3270-style keyboard assist you in modifying data on the screen. Some have equivalents on workstation terminals while others do not. These keys and their generic key sequences are

Key Name	Key Sequence	Function
Insert	CTL a	enter terminal insert mode
Reset	CTL r	terminate insert mode
Delete	CTL d	delete one character
Erase-EOF	CTL e	delete to end of input field

In many cases, the INSERT and DELETE keys on your terminal keyboard are automatically recognized by uni-SPF.

Few, if any, terminals have keys equivalent to the 3270-style RESET and ERASE-EOF. If there are keys on your keyboard that are not in use for other uni-SPF functions, you may choose to map them to one or more of the hardware edit keys. A frequent choice for ERASE-EOF is the key marked "END". The "Keybind Maintenance Utility" section in ***Chapter 3: Parameters*** guides you in using the keyboard test and mapping utility, uni-KEY, to perform such keyboard mappings. uni-KEY is accessed through Option 0.K.2.

Unlike PC and workstation style keys, the 3270-style INSERT is not a toggle. When you press INSERT, a caret (^) appears near the center of the last line on the screen. This alerts you that you are in insert mode and that characters typed on the keyboard will be inserted into the text at the current cursor position. Press the RESET key to terminate insert mode. The caret indicator disappears when insert mode is terminated. Pressing Enter or a PF key automatically terminates insert mode.

Special Key

There is one key that performs a special function with-in uni-SPF. It inserts a non-printing tab character into the text at the cursor position. This tab character is a hexadecimal `09' which is processed by the editor to position text in expanded mode. The generic key sequence to insert a tab character in the text is CTL-T. If you use this feature frequently, you may choose to map a single key to it. Refer to the sections on keyboard mapping in ***Chapter 3: Parameters*** for assistance.

Keyboard Mapping

uni-SPF uses the Unix System V curses library for terminal input and output. It determines your terminal type from the TERM environment variable and uses the corresponding terminfo database entry for keyboard mapping. In some cases, the terminfo database provided by the workstation manufacturer may not contain definitions for all the keys used by uni-SPF; or the database may contain incorrect definitions for some keys.

In other cases, there may be extra keys on the keyboard that you would like to assign to a specific function. uni-SPF includes a utility, uni-KEY, that allows you to overcome terminfo deficiencies or use extra keys by mapping a specific key to a uni-SPF keyboard function through a special facility called KEYBIND. Option 0.K.1 displays the keybinds that are currently in effect and allows you to modify the current definitions. Option 0.K.2 executes the uni-KEY utility, through which you

can add or change keybinds. The section "Keybind Maintenance Utility" in ***Chapter 3: Parameters*** provides details on the use of the uni-KEY utility.

KEYBIND is the recommended method of mapping keyboard definitions for use with uni-SPF since it allows custom keyboard definitions for each user on each workstation/terminal combination. Other methods (discussed briefly) may create incompatibilities in keyboard mappings for other applications in use at your site.

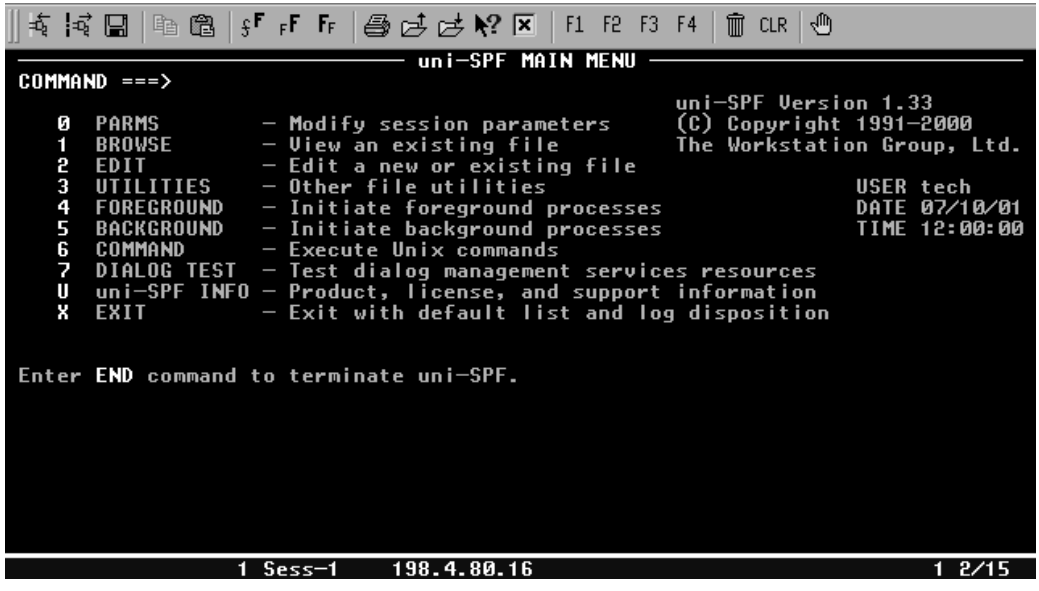
The alternate methods to map keyboard sequences are changing the master terminfo database for all users or creating a local definition that is accessed through the environment variable TERMINFO. If you choose to do this, the table below shows the terminfo keynames that uni-SPF uses for each function. You must be sure that the entry for each terminal type to be used by uni-SPF has all these key names included and has the actual key sequence that the terminal transmits as the definition for that key name.

Keyboard Function	**Terminfo Keyname**
up arrow	kcuu1
down arrow	kcud1
left arrow	kcub1
right arrow	kcuf1
function keys 1-24	kf1, kf2, kf3, ..., kf24
home	khome
delete	kdch1
insert	kich1
reset	krmir
erase-eof	kend

In some terminfo definitions, kf0 is a synonym for kf10. uni-SPF recognizes either keyname.

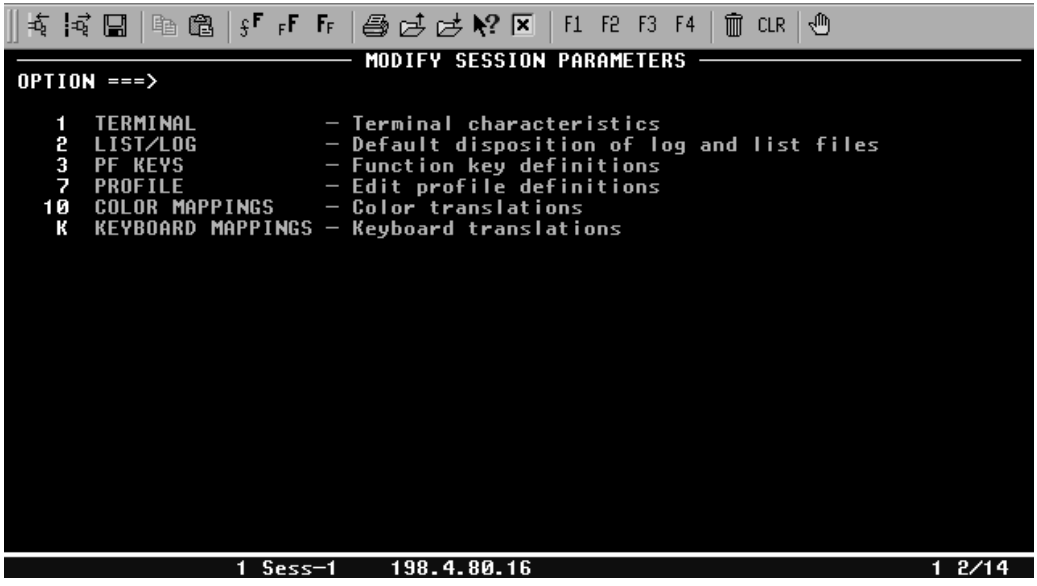
uni-SPF Panels

User interaction with uni-SPF is accomplished through a series of panels. One type of panel is simply a menu for selection of options available at the current level in the hierarchical structure. The uni-SPF Main Menu panel is an example of this type of panel. On such panels there is only one input field, indicated by COMMAND ===> or OPTION ===>, and the cursor is automatically positioned here when the panel appears.



The only input you provide is the number or letter corresponding to your choice of option. When you press Enter, the panel associated with that option appears.

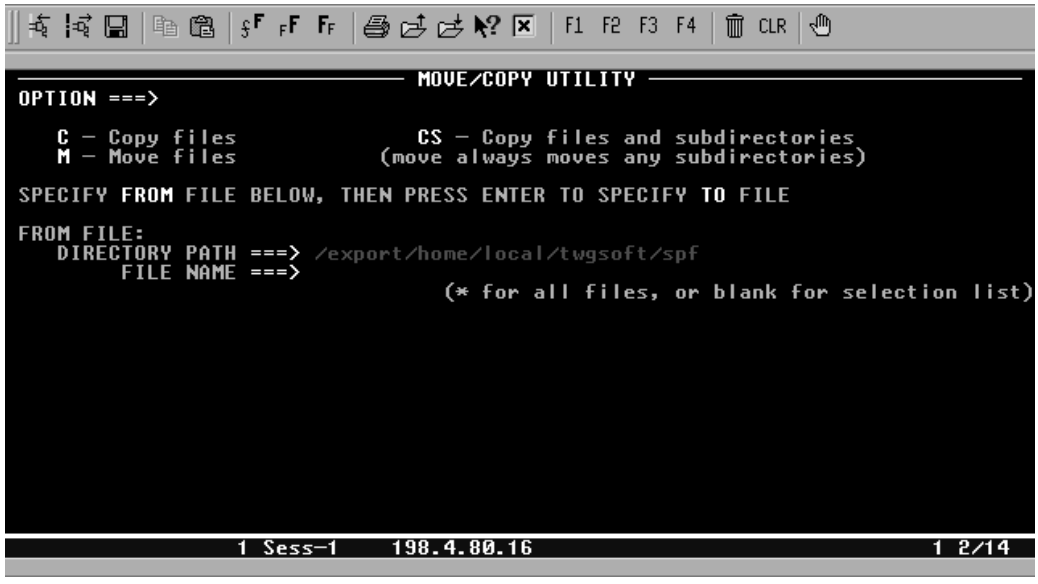
Another example of an option panel is the uni-SPF Modify Session Parameters panel, which appears when you select 0 (PARMS) from the Main Menu panel.



This panel shows the secondary options available at this level.

A second type of panel is the entry panel, which displays input fields where you type the information necessary to complete a command or process an option. The primary input field may be indicated by OPTION ===> or COMMAND ===>.

Other input fields may be blank, or they may contain current information from system defaults or from the most recent use of an option. The Move/Copy Utility and the Edit-Entry Panel are examples of this type.


F1 F2 F3 F4 CLR
MOVE/COPY UTILITY
OPTION ===>
C - Copy files
M - Move files
CS - Copy files and subdirectories
(move always moves any subdirectories)
SPECIFY FROM FILE BELOW, THEN PRESS ENTER TO SPECIFY TO FILE
FROM FILE:
DIRECTORY PATH ===> /export/home/local/twgsoft/spf
FILE NAME ===>
(* for all files, or blank for selection list)
1 Sess-1 198.4.80.16 1 2/14


F1 F2 F3 F4 CLR
EDIT - ENTRY PANEL
COMMAND ===>
FILE SPECIFICATION:
DIRECTORY PATH ===> /export/home/jac/bin
FILE NAME ===> regrep.rex
(blank for selection list)
EDITOR ===> SPF
(SPF, XEDIT, or VI)
PROFILE NAME ===>
(Blank defaults to type)
INITIAL MACRO ===>
1 Sess-1 198.4.80.16 1 6/25

For these panels, the initial cursor position may be the primary input field, as for the Move/Copy Utility; or it may be the field required to invoke the requested process, as in the Edit-Entry Panel where a filename must be supplied for the Editor to begin.

A third type of panel is the scrollable list, of which the file selection panel is the most common. Whenever an entry panel requires a file name as input, you may

- accept the current names provided by the system
- change both the directory path and the file name
- specify a file name that contains wild-card characters
- leave the file name blank

The first time you use uni-SPF, it supplies a "." for the directory path to indicate the current working directory. Subsequently, the directory path is taken from the name most recently provided in an entry panel.

If the filename is blank or contains wild-card characters, uni-SPF displays a file selection panel. When the filename is blank, this panel contains a listing of all files in the indicated directory. When the filename contains wild-card characters, the file selection panel contains a listing of all files that match the filename template.

When a file selection panel appears, the tab key moves to the input fields associated with each filename. The command line at the top of the panel shows the commands that are valid in the input area to the left of the filename. In most file selection panels, typing "s" to select a file for the current operation is the most common entry.

File selection panels also identify the type of file. In the column labelled "T", uni-SPF identifies the following types:

d	directory
l	link
s	socket
f	named pipe (fifo)
c	character special device
b	block special device

You may change to another directory by typing “s” in the input area to the left of the desired directory name.

For specifying filenames, uni-SPF supports the following standard Unix wild card conventions:

*	matches any group of characters
?	matches any single character
[ ]	contains a list of acceptable matches for a single character

These wild card characters may be used singly or in combination to form the appropriate filename pattern. The following table gives examples of how Unix matches wild card characters for file names:

Filename	**Matches**
*	all files in the directory
.	all files in the directory that have an extension; matches myfile.dat and yourfile.text but not myfile or yourfile
sc*	all files in the directory that begin with sc; matches schedule, scan, scan.file, and scan.for.text.
sc*.*	all files in the directory that begin with sc and have an extension; matches scan.file and scan.for.text but not schedule or scan
list?	all files in the directory with 5-character names that begin with “list” and that do not have an extension; matches list1, list2, and list3 but not list1.a, list1.b, or list1.backup

list?.*	matches all files in the directory with 5-character names that begin with “list” and that have an extension; matches list1.a, list1.b, and list1.backup but not list1, list2, or list3
list?.?	all files in the directory with 5-character names that begin with “list” and that have a 1-character extension; matches list1.a and list1.b but not list1, list2, list3, or list1.backup
l[ia]st	all files with 4-character names where the second character may be either “i” or “a”, the other characters as shown, and no extension; matches list and last but not lost or list.today
l[iao]st.*	all files with 4-character names where the second character may be “i”, “a”, or “o”, the other characters as shown, and any extension; matches list.today but not list, last, or lost
l[io]s?*	all files with names where the first character is “l”, the second character may be “i” or “o”, the third character is “s”, the fourth character may be any character, and any number of characters, including an extension, may follow; matches list, listing, listing.file, lisp.program, lost, losalt.diet, and lost.and.found but not last or lesson
l[a-f]s?*	same as the previous pattern except that the second character may be any character from “a” through “f”; matches last and lesson but not list, listing, listing.file, or lost

Color Support

uni-SPF provides support for the use of color in panel displays. This support is subject to the following restrictions:

- the specific Unix implementation must support System VR4 curses or later
- the terminal hardware must be capable of exhibiting the behavior
- the terminfo entry for the specific terminal must define an escape sequence for the behavior

Default colors are as follows:

Field Type	Intensity	Default Color
Text/Output	High	White
Text/Output	Low	Turquoise
Input	High	Red
Input	Low	Green

The environment variable SPFCOLOR controls the display of colors. Set SPFCOLOR to ON to enable color display; this is the default, even if the environment variable is unset. Set SPFCOLOR to OFF to disable color display. SPFCOLOR may be set by an individual user or it may be set in the "ispf" startup script for all users.

Dynamic Resizing of Display

uni-SPF correctly responds to window resizing that occurs after the session is started. In some cases, it may be necessary to press Enter to correctly redisplay text or output fields after the resize occurs. This feature applies only to sessions of uni-SPF running in a window in an X-windows environment such as Motif or the Common Desktop Environment (CDE).

uni-SPF Info (Option U)

The uni-SPF Main Menu panel includes a selection to access product, license, and support information. The information available includes

- Product Information, including the version being executed, the platform type on which it is running, and the hostname of the workstation or server on which it is running
- License Information, including software serial number, maximum concurrent users, and additional information that might be requested for analysis of technical problems
- Technical Support information, including telephone, FAX, email, and World-Wide Web contact information for TWG support staff

File Sizes for Edit and Browse

uni-SPF supports the following file characteristics for Edit and Browse:

	Browse	Edit
Maximum record length	32,768	4,096
Maximum number of lines	10,000,000	999,999

The limit of 999,999 lines in Edit is the result of having only 6 character positions for line numbers in the prefix area. The maximum number of lines may be further limited by the amount of memory and/or swap space available on the workstation or server being used.

Navigating the System

When you start uni-SPF, the first panel that appears is the Main Menu panel. This is a menu listing all the features available in the system. Select the desired option by typing its number or letter in the primary input field and pressing Enter. This takes you to the initial panel for that option. When the primary option you choose has secondary options, the initial panel for that primary option selection is also an option panel from which you choose the desired secondary option.

Chapter 1: Introduction contains a discussion of the uni-SPF hierarchy and a chart of the options at each level. Chapters 3 through 10 provide detailed reference for Main Menu options 0 through 7. Each chapter includes illustrations of the main panel for that option and of secondary panels that may appear.

Provide the requested input on any panel to proceed to the next panel. From any panel, type “END” in the primary input field or press PF3 to return to the previous panel. To end the current session, return to the Main Menu panel and select “X”, type “END”, or press PF3. You may move immediately to the Main Menu panel from any other panel in the system by typing

"RETURN" in the primary input field or by pressing PF4.

Frequent users will soon know exactly which panel they need to accomplish the task at hand. The "=" command allows you to move from any panel in the system directly to any other panel by specifying the desired panel number. Typing "=3.3" in any field of any panel takes you immediately to the Move/Copy Utility panel. Typing "=2" in any field of any panel takes you immediately to the Edit-Entry panel. Typing "=x" in any field of any panel immediately ends the current uni-SPF session. (This does not apply to scrollable fields.)

Panel Input

Each panel has one or more input fields. The primary input field on each panel follows the OPTION ===> or COMMAND ===> indicator. On an option panel, this is the only input field on the panel. Although the arrow keys will move the position of the cursor, input is not accepted anywhere else on the screen.

When a panel has more than one input field, the tab key moves the cursor automatically to the beginning of the next input field. Back-tab moves the cursor to the beginning of the previous input field. You may also use the arrow keys to move the cursor, but you must then be sure that you are in a valid input field before typing your entry.

Panel Scrolling

Some panels may contain more information than can be displayed on a single screen. File selection panels are a typical example. In these cases, you may need to scroll the screen to see the information you need and its associated input fields.

Whenever a panel contains more than one screen of data, the scroll indicator (SCROLL ==>) appears in the upper right corner of the screen. Scrolling is permitted whenever this indicator is present. The uni-SPF commands UP and DOWN control scrolling of the screen.

These commands are normally assigned to PF keys as follows:

UP	PF7
DOWN	PF8

You may type the desired scrolling command in the primary input field or press the appropriate PF key to scroll the panel.

These commands have several operands that control the amount of movement in the direction indicated. These operands are discussed fully in the section “uni-SPF Commands” in this chapter. The field that follows the scroll indicator (SCROLL ==>) contains one of the operands of the scrolling commands. This is the operand used by default when you press a scrolling PF key or type the command without an operand. The system default for this field is PAGE, which causes scrolling to move a full screen at a time. Because this field is an input field, you may move the cursor there and type any other valid scrolling operand to change the current scrolling default. If you change the value in the scrolling indicator to MAX, it reverts to the previous setting as soon as you press a PF key or Enter.

To override the default scrolling value

- type a scrolling command with a different operand on the command line, or
- type the desired operand on the command line and press a scrolling PF key

Operands of the scrolling commands are documented as part of the command documentation in the next section of this chapter.

uni-SPF Commands

There are a number of uni-SPF commands that may be entered on any panel. Each of these commands is fully documented here. These commands are also valid in options such as Edit and Browse, which have their own command set. For convenience, they are also documented in those chapters.

CURSOR

The CURSOR command moves the cursor to the prim-ary input field on the current panel. Its function is the same as the HOME key.

CURSOR

Since this command is meaningless when issued from the primary input field, it is normally used by assigning it to a PF key or mapping it to another keyboard sequence.

DOWN

The DOWN command moves the display toward the bottom of the file. It is normally associated with the PF8 key.

DOWN [Page]
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, DOWN moves toward the bottom of the file by the amount specified in the SCROLL field in the upper right corner of the screen. The section in this chapter entitled "Panel Scrolling" contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the DOWN command to override the SCROLL field value.

You may use these operands in conjunction with a PF key by typing the operand on the command line and pressing the PF key associated with DOWN (normally, PF8).

Examples:

`DOWN`	scrolls toward the bottom of the file by the amount indicated in the SCROLL field
`DOWN 12`	scrolls down 12 lines
`DOWN M`	scrolls to the bottom of the file

END

The END command terminates the current panel and returns to the previous panel. It is normally associated with the PF3 key.

END

When issued from the Main Menu panel, this command terminates the uni-SPF session. When END is issued from Edit, uni-SPF uses the AUTOSAVE setting to determine whether to save changes to the file.

HELP

The help command displays a long message providing additional details about the short message that is currently on the screen. It is normally associated with the PF1 key.

HELP

Example:

If you type "Q" as a selection on the Main Menu panel, the message

```
Invalid option
```

appears in the message area of the screen.

If you then type "HELP" or press PF1, the message

```
You have entered an unrecognized command or
option in the command field
```

appears.

LOCATE

The LOCATE command is available for use on File Selection panels. It repositions the display with the specified file at the top of the panel. LOCATE therefore provides directed scrolling for these panels.

LOCATE ***filename***
LOC
L

LOCATE searches for ***filename*** in the file list and scrolls the display to position the entry for ***filename*** at the top.

If an identical match for ***filename*** is not found, LOCATE finds the nearest partial match. When a partial match is found, that entry is positioned next to the top of the display. When no match is found, the display is positioned at the row immediately above where ***filename*** would normally occur in the list.

This form of the LOCATE command is valid on all File Selection panels. However, the command has additional syntax and different action when used in the Directory Utility (Option 3.1). This specific behavior is documented in ***Chapter 6: Utilities***.

PFSHOW

The PFSHOW command controls the display of PF key definitions at the bottom of the screen.

PFSHOW [ON]
OFF
[TAILOR]

PFSHOW with no operands is identical to PFSHOW ON. This causes PF key definitions to appear on the last two rows of the screen.

PFSHOW OFF removes the PF key definitions from the display.

PFSHOW TAILOR allows you to define the display that occurs when PFSHOW ON is executed. On the "Tailor PF Key Definition Display" panel, you may define which is the primary set of PF keys (PF1-12 or PF13-24) and which set appears when you type PFSHOW ON.

REFRESH The REFRESH command redisplays the current screen in its entirety. This is useful if the display has been corrupted by an error condition or operating system console messages.

REFRESH

RETRIEVE The RETRIEVE command retrieves the last command executed from the command buffer and redisplays it on the command line. It is normally associated with the PF12 key.

RETRIEVE

uni-SPF maintains a buffer of commands that have been executed. Press PF12 repeatedly to retrieve earlier commands from the buffer.

RETURN The RETURN command returns directly to the uni-SPF Main Menu panel from any panel in the system. It is normally associated with the PF4 key.

RETURN

When RETURN is issued from Edit, uni-SPF uses the AUTOSAVE setting to determine whether to save changes to the file.

SELECT

The SELECT command allows you to select a file from a File Selection panel or to change the current directory in a File Selection panel.

SELECT *name*
SEL
S

If ***name*** is a directory, uni-SPF changes the current directory to the one specified.

If ***name*** is a file, the command selects this file for the current operation. ***name*** may be a full path name to select a file in the current directory or in a different location.

This form of the SELECT command is valid on any File Selection panel. Extensions to the use of SELECT may be available within some uni-SPF options. Such extensions are fully documented where they apply.

SHELL

The SHELL command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The SHELL command is synonymous with the ! command.

SHELL [*string*]

With no operands, SHELL takes you to the Unix shell. The shell chosen is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press Enter when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. The message

```
Press ENTER to continue uni-SPF
```

appears after the output of the Unix command.

SPLIT

The SPLIT command splits the screen at the current cursor location and starts a second uni-SPF task. It is normally associated with the PF2 key.

SPLIT

After the SPLIT command, the new uni-SPF task becomes the current session. The active session is identified by the cursor position.

To switch between the sessions (or screens) use the SWAP command or its equivalent PF key (normally PF9); or move the cursor to the desired session. The END command (or PF3) from the Main Menu panel or "=x" from any other panel ends the current session and returns the screen to its original configuration.

If you enter SPLIT or press its equivalent PF key when there are already two sessions displayed, the new SPLIT command re-splits the screen at the current cursor location. It does not change the display in either window.

SWAP

The SWAP command toggles between uni-SPF sessions. It is normally associated with the PF9 key.

SWAP

For sessions activated with SPLIT, it changes the active session from one portion of a split screen to the other by re-positioning the cursor.

UP

The UP command moves the display toward the top of the file. It is normally associated with the PF7 key.

UP **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, UP moves toward the top of the file by the amount specified in the SCROLL field in the upper right corner of the screen. The section in this chapter entitled “Panel Scrolling” contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the UP command to override the SCROLL field value.

You may use these operands in conjunction with a PF key by typing the operand on the command line and pressing the PF key associated with UP (normally, PF7).

Examples:

`UP`	scrolls toward the top of the file by the amount indicated in the SCROLL field
`UP 12`	scrolls up 12 lines
`UP M`	scrolls to the top of the file

! (Exclamation Point)

The ! command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The ! command is synonymous with the SHELL command.

![*string*]

With no operands, ! takes you to the Unix shell. The shell chosen is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press ENTER when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. The message

```
Press ENTER to continue uni-SPF
```

appears after the output of the Unix command.

= (Jump)

The = command invokes the "jump" feature. This allows you to move directly from any uni-SPF panel to any other uni-SPF panel simply by specifying the panel number.

=*number*

number may be any valid uni-SPF option number. If it is a single digit, it refers to the entries on the Main Menu panel. Specify secondary options by typing the primary option, a period, and the secondary option number as in 3.3 for the Copy/Move Utility. You may enter an = command in any input field of any panel.

Examples:

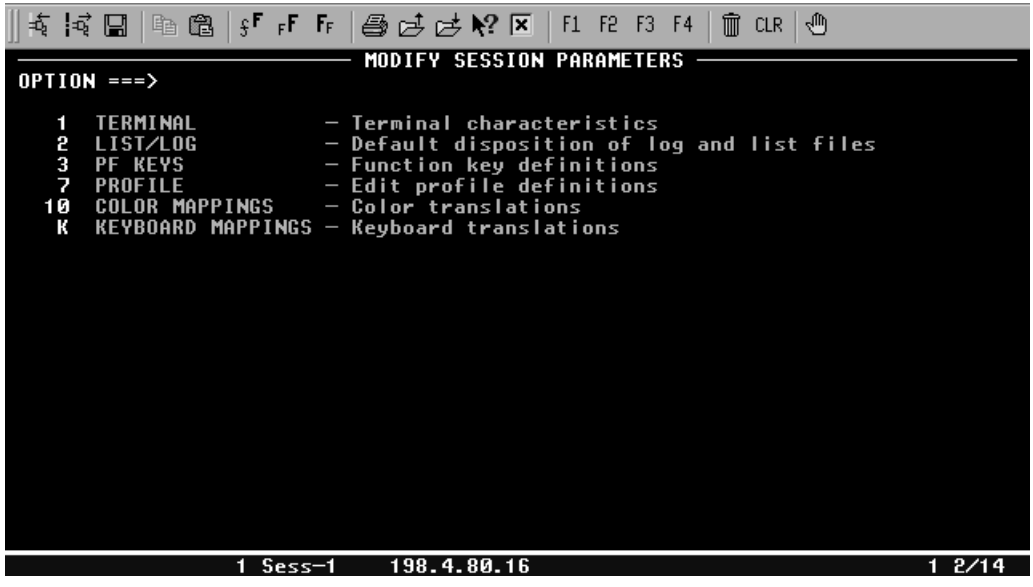
`=3.1`	moves immediately to the directory utility from any other panel in uni-SPF
`=2`	moves immediately to the Edit-Entry panel from any other panel in the system

Chapter 3: Parameters

The uni-SPF Modify Session Parameters feature (Option 0) allows you to establish default settings for certain configuration options. Specifically, you may

- set terminal display options (Option 0.1)
- define a procedure for routing printed output in your environment and for processing log files (Option 0.2)
- modify the definition of Program Function (PF) keys (Option 0.3)
- create, modify, or delete editor profiles (Option 0.7)
- view or define keyboard mappings (Option 0.K)

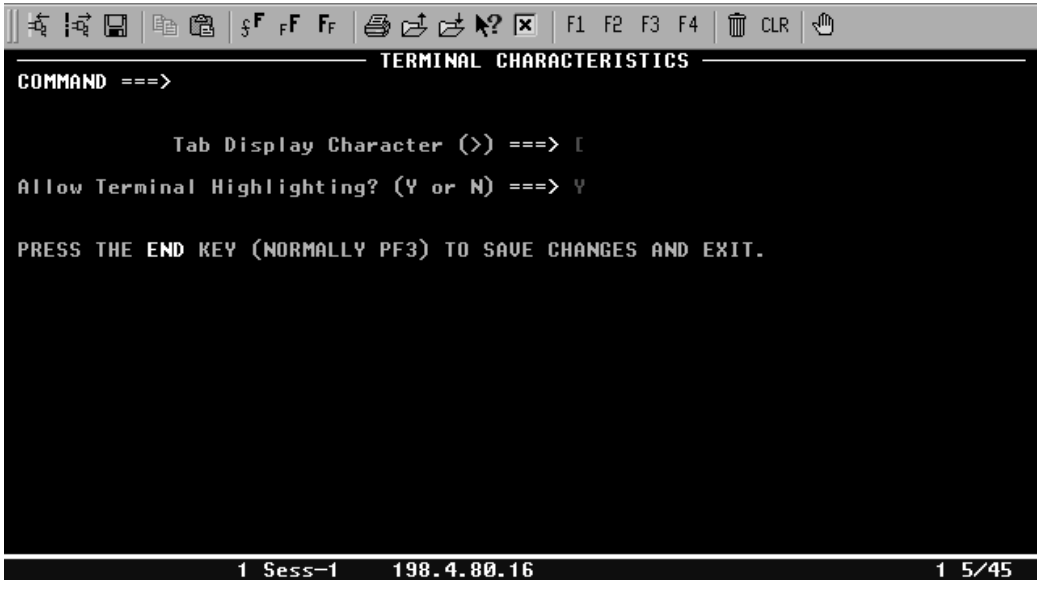
The uni-SPF Modify Session Parameters panel is a menu from which you select the parameter to modify. It is shown on the following page.



Select the desired option this panel to continue with parameter modifications.

Terminal (Option 0.1)

This option allows you to set certain terminal display characteristics:



Tab Display Character

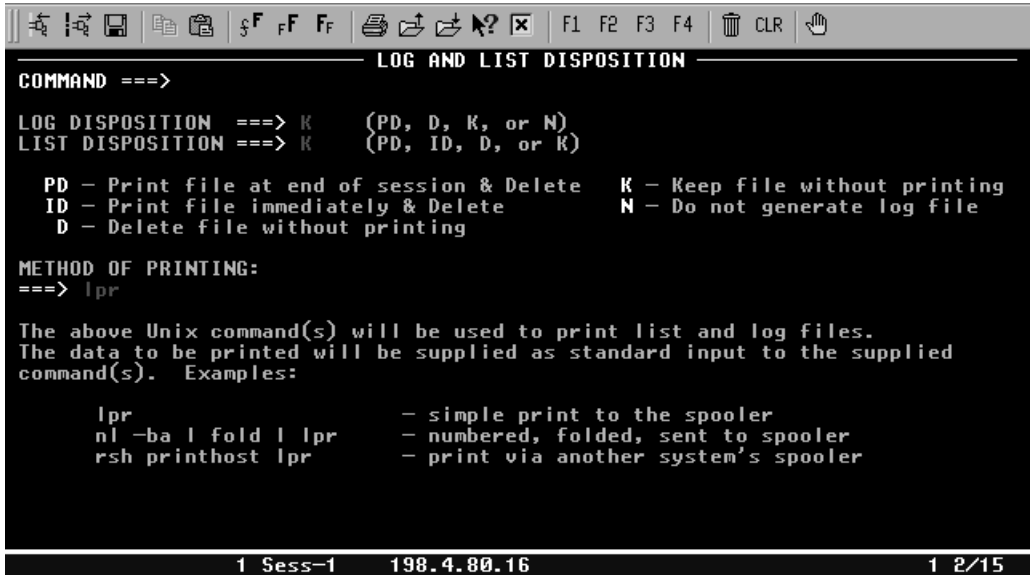
initial default is ">"; you may set this to any character, including blank

Allow Terminal Highlighting

initial value is "Y"; set this value to "N" for terminals on which highlighting or reverse video produces undesirable results

List/Log (Option 0.2)

The List/Log panel allows you to specify the procedure for routing printed output to the desired printer in your environment. This may be a command, a shell script, or a pipe for which you supply a file name. uni-SPF panels that support a print command will automatically use this procedure for printing.



The Log Disposition specifies the disposition of the uni-SPF log file when you exit a session. Processing options are shown in parentheses, with the definitions provided below. The default for LOG processing is K. When this choice is in effect, the log file is saved in your HOME directory.

The List Disposition specifies the disposition of list files created during this uni-SPF session. This includes files selected for printing under the Utilities feature (Option 3). Processing options are shown in parentheses, with the definitions provided below. The default for LIST processing is K. When this choice is in ef-

fect, the list output is placed in your HOME directory when you terminate the uni-SPF session.

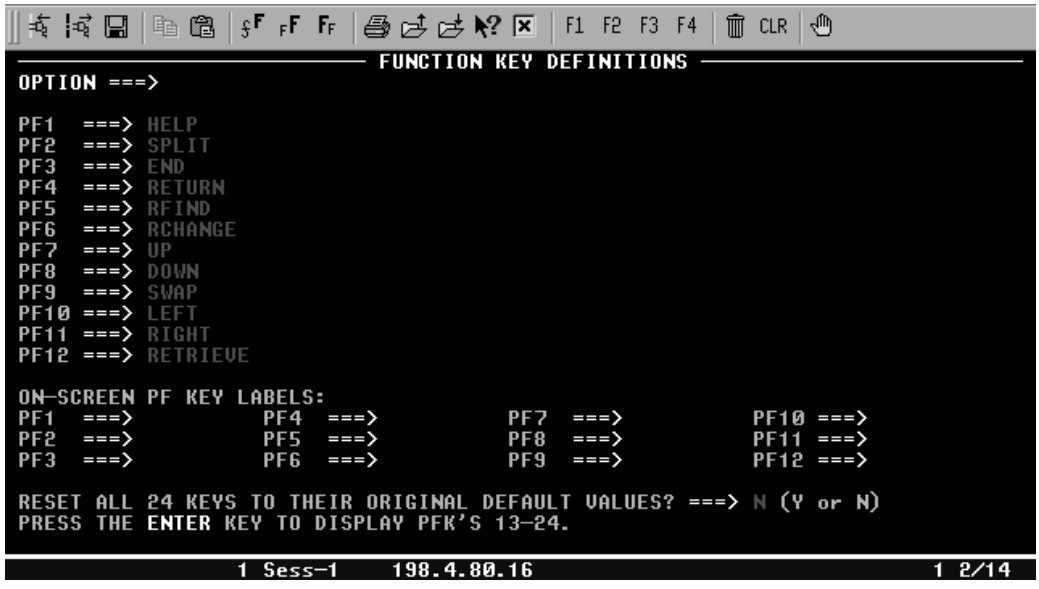
The Method of Printing field allows you to specify the correct method for routing printed output on your system. Specify the Unix command(s) that you normally use to print a file.

PF Keys (Option 0.3)

The Function Key Definitions panel displays your current PF key definitions and allows you to modify them. uni-SPF supports twenty-four PF keys, with the initial definitions for PF13-24 identical to those for PF1-12. These initial settings are shown below.

PF1	PF13	HELP
PF2	PF14	SPLIT
PF3	PF15	END
PF4	PF16	RETURN
PF5	PF17	RFIND
PF6	PF18	RCHANGE
PF7	PF19	UP
PF8	PF20	DOWN
PF9	PF21	SWAP
PF10	PF22	LEFT
PF11	PF23	RIGHT
PF12	PF24	RETRIEVE

The primary Function Key Definition panel displays the current settings for PF1-12.



The tab key moves the cursor automatically to the beginning of the input field for each PF key. Press Enter to view the secondary Function Key Definition panel, where the current settings for PF13-24 appear.

The Enter key functions as a toggle between the two Function Key Definition panels. Press PF3 from either panel to exit Function Key Definitions.

To change the definition of a PF key, move the cursor to its input field and type the new definition over the old. To remove a definition, leaving that PF key with no assigned function, simply erase the existing definition.

The Function Key Definitions panels include fields for setting labels that appear when PFSHOW ON is set to display PF key definitions on the screen. There is also an option for restoring all PF key definitions to their initial default values.

To save new definitions, press PF3. To discard modifications, you **must** retype the desired PF key definitions before you press PF3.

You may define PF keys to perform

- basic uni-SPF commands, including Browse or Edit commands
- a sequence of commands that constitute an operation you use frequently
- Edit line commands

Initial PF key definitions are basic uni-SPF commands. To define a PF key to perform a sequence of commands, use the semi-colon line terminator to separate individual commands. As an example, if you define PF12 as

```
SORT;BOUNDS 1 72;DOWN MAX
```

then while you are in the Editor, you can use a single keystroke (PF12) to perform this command sequence.

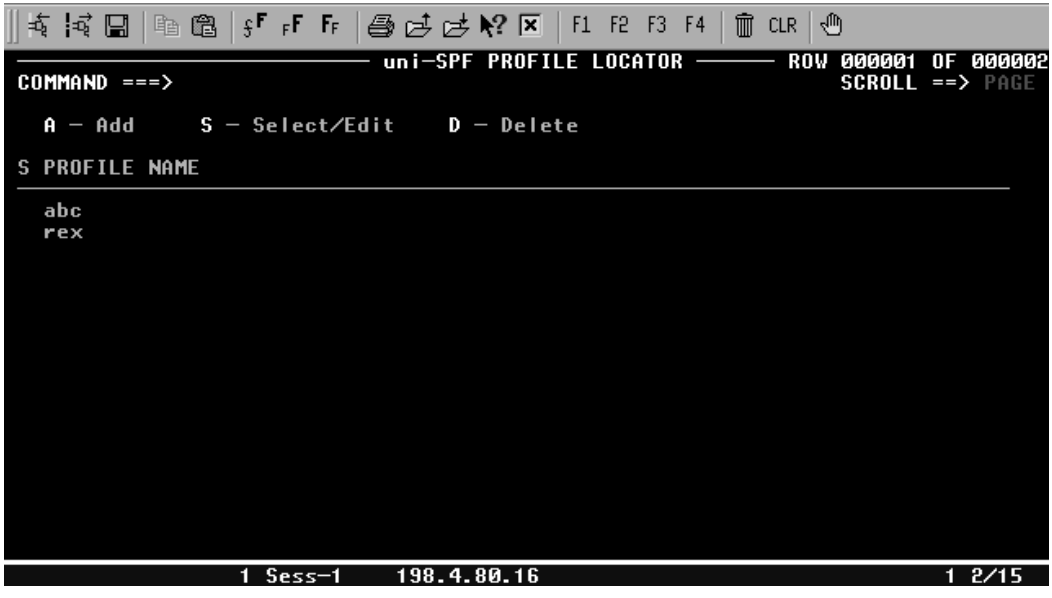
To define a PF key to perform an Edit line command, you must begin the definition with a colon. Thus if you define a PF key as

```
:TS
```

you may use that PF key to perform the Text Split line command.

Profile (Option 0.7)

The uni-SPF Profile Locator panel allows you to add, modify, or delete edit profiles.



To add a new profile entry, type

```
a prof-name
```

in the COMMAND field. ***prof-name*** is the name assigned to the new profile.

To select an existing profile for review or modification, type

```
s prof-name
```

in the COMMAND field or type `s` in the “S” field of the panel display.

To delete a profile, type

```
d prof-name
```

in the COMMAND field or type `d` in the "S" field of the panel display.

If you add a new profile or select an existing profile, the uni-SPF Profile Editor panel appears as shown below.

You may set any of the editor profile values in this panel. Three of the fields do not have corresponding editor primary commands and **must** be set in this panel.

RECORD FORMAT

Use DATA for files that have normal line-feed characters at the end of each line of data.

Use LENGTH for files that do not have normal line-feed characters; when you edit a file using a

length-delimited profile, uni-SPF displays the file with the specified number of characters on each line; if you save the file, uni-SPF automatically restores the file to its original configuration before saving.

MAX RECORD LENGTH

For length-delimited file displays, this is the number of characters that will appear on each line of the edit display.

For all file displays, this is the maximum width of a data line that may be entered; it is also the limit for a scrolling operation to the right.

DISPLAY CHARACTER SET

This field controls the display output for files. The default is ASCII. If you edit a file that contains EBCDIC data, the standard ASCII editor display will treat the EBCDIC data as "non-printable".

Set this field in the profile to "EBCDIC" if you wish to see the character representation of the EBCDIC data.

This field affects **only** the display. The actual data is retained in EBCDIC. The display resulting from HEX ON shows the EBCDIC hexadecimal values for the data. Changes to the file are saved in EBCDIC.

Keyboard Mappings (Option 0.K)

Chapter 2: Interaction with uni-SPF contains a complete discussion of the purpose of and methods for keyboard mapping in the section entitled "Keyboard Mapping". The keyboard mapping options allow you to

- view or modify keyboard mappings that are currently in your keybind table
- define new keyboard mappings
- import or export standard keyboard mapping sets

The Keyboard Mappings option panel is a menu from which you choose to view or define keybinds.

Select 1 (Option 0.K.1) to view or modify your current keybinds. Select 2 (Option 0.K.2) for the Keybind Maintenance Utility that allows you to test keys and to define new keybinds. This option invokes the keyboard mapping utility, uni-KEY, to create new keybind definitions. Select 3 (Option 0.K.3) to import or export standard keybind sets.

Keybind Display/Update (Option 0.K.1)

When you select this option, the Keybind Display/Update panel appears. It shows a listing of all keybinds that are currently in your keybind table (ISPKBND.tbl in your HOME directory). This provides a quick reference for keyboard mappings for any terminal that you use.

The fields in the display are

Terminal the terminal type for which this keybind definition is valid. This corresponds to the TERM environment variable.

Key Description the description of the related keyboard key as it is actually printed on the key on this terminal's keyboard.

Sequence the key code that is transmitted by this key. If you do not know the keycode,

the Keybind Maintenance Utility (Option 0.K.2) provides an option to determine codes transmitted by individual keys.

Function the uni-SPF keyboard function that this key activates.

The tab key moves automatically to each field in the keybind definition for input. Use the scrolling commands or their corresponding PF keys to reposition the display as needed (***Chapter 2: Interaction with uni-SPF*** contains a detailed description of the scrolling commands and features.)

To define a new keybind, type “i” in the primary input field to insert a new keybind record. The terminal type is automatically taken from the current setting of the TERM environment variable.

To modify an existing keybind definition, tab to the field that you wish to change and type the revised data.

To remove a keybind definition, tab to the input field immediately to the left of the desired record and type “d”. You may delete several records simultaneously by typing “d” before each record to be removed.

Keybind definitions and modifications made through the Keybind Display/Update panel are effective immediately. If you are only adding new keybinds, it is not necessary to exit and re-start uni-SPF. If you are deleting keybinds or replacing old definitions with new ones, we recommended that you exit and re-start uni-SPF.

Keybind Maintenance Utility (Option 0.K.2)

The Keybind Maintenance Utility automatically invokes uni-KEY, the keyboard mapping utility from The Workstation Group. It allows you to determine what keycodes are transmitted by individual keys on your terminal keyboard. It also provides another method for defining new keybinds for use with uni-SPF. Keybinds defined using the Keybind Maintenance Utility are automatically stored in your keybind table and are effective immediately.

uni-KEY reads your current keybind table (the file ISPKBND.tbl in your HOME directory). If you choose to define new keyboard mappings, uni-KEY automatically updates this table.

The uni-KEY main menu is shown below.

```
uni-KEY:  curses/TERMINFO keyboard utility
 Please choose a function:
  t - test special keys
  u - uni-SPF keybind maintenance
  x - exit
```

From this menu you may select one of uni-KEY's two functions or press "x" to exit the program. Once you are in the "t" or "u" functions, you may press ctl-x (when that option appears) to end that function and return to the main menu.

Testing Special Keys

To determine what keycodes are transmitted by individual keys, select "t". The screen display changes to

```
uni-KEY:  curses/TERMINFO keyboard utility
 test - keyboard test utility - ctl-x to end
```

and the cursor is positioned on the first blank line of the screen. Press any key or a key combination such as control in conjunction with a key. uni-KEY echoes the keycode transmitted. If the key you pressed is already mapped by curses and terminfo, that definition appears on the screen.

uni-KEY uses a timing scheme to determine when you have completed a key sequence. Allow a few seconds between key sequences. uni-KEY echoes one keycode per line on the screen. If you press keys too fast, uni-KEY cannot distinguish the end of one key sequence and the beginning of the next. When this happens, all keycodes appear on the same line until you pause briefly between key sequences.

When a key sends an unprintable character, uni-KEY uses the standard notation conventions to represent these codes. As an example, Escape is shown as \e. The keycodes echoed to the screen can be used exactly as they appear in Option 0.K.1 or to update a terminfo definition.

Defining Keybinds

If you select “u” from the uni-KEY main menu, the following display appears:

```
uni-KEY:  curses/TERMINFO keyboard utility
 Choose a key with the cursor and press ENTER,
 (or press ctl-x to return to the main menu.

 -(press enter here if your arrow keys don't work,
   and you can type the name of the key you want)

ARROW KEYS  FUNCTION KEYS  OTHER KEYS

 -UP         -PF1  -PF13    -BACKSPACE
 -DOWN       -PF2  -PF14    -BACKTAB (Move Cursor to Previous Field)
 -LEFT       -PF3  -PF15    -DELETE  (Delete Character)
 -RIGHT      -PF4  -PF16    -END     (Erase EOF)
             -PF5  -PF17    -ENTER
             -PF6  -PF18    -HOME
             -PF7  -PF19    -INSERT  (Set Insert Mode)
             -PF8  -PF20    -PGDN    (Scroll One Screen Down)
             -PF9  -PF21    -PGUP    (Scroll One Screen Up)
             -PF10 -PF22    -RESET   (Reset Insert Mode)
             -PF11 -PF23    -TABCHAR (Place a TAB Character in the File)
             -PF12 -PF24    -TABKEY  (Move Cursor to Next Field)
```

Use arrow keys to move the cursor to the name of the key for which you wish to create a keybind. Press Enter. uni-KEY prompts you to press the key you want to associate with this function.

uni-KEY then prompts you for the name of the key that you pressed. This is the descriptive information that identifies which keyboard key is related to this function.

After you have answered all the prompts, the key name selection menu re-appears so that you may define more keybinds or terminate this mode.

If your arrow keys do not function, press Enter immediately from the keyname menu. uni-KEY prompts for the name of the desired key. Type the key name as it appears on the menu (but without the leading dash) and proceed as described above.

Some systems do not support the timing feature that allows uni-KEY to determine when a complete key sequence has been received. On such systems,

uni-KEY prompts you to type ctl-x after you have completed your keyboard sequence. Other prompts proceed as described above.

uni-KEY determines your current terminal type from the environment variable TERM. This terminal type is associated with each keybind in the keybind table.

When you define new keybinds, uni-KEY checks to determine if you have an existing keybind for the keyname/key-sequence/terminal-type combination that you are now attempting to set.

If a match is found, uni-KEY takes one of the following actions:

- **Key-name/Terminal-type Match:** If the key name you chose from the menu is already bound to another key for this terminal type, uni-KEY warns you of the duplication. You may delete the existing keybind and replace it with a new one or retain the existing keybind.

  If you choose to retain the existing keybind, you may then bind another key to the same key name or press ctl-x to keep a single definition.

- **Key-sequence/Terminal-type Match:** If the key you pressed is already bound to a different key-name for this terminal type, uni-KEY prompts you to select another key. At the present time, there is no support in uni-KEY for removing such keybinds.

uni-KEY Limitations

This current version of uni-KEY has the following limitations:

- No more than 100 keybinds mayt be defined. Existing keybinds in the table and new keybinds created in this uni-KEY session are added to determine the total.
- The individual key name or command to be bound to a key sequence may not exceed 80 characters.
- The key sequence (keycode) transmitted by the individual key may not exceed 20 characters.

- A key description (for the comments) may not exceed 50 characters.
- The terminal name (from the environment variable TERM) may not exceed 20 characters.

Keybind Import/Export (Option 0.K.3)

uni-SPF also provides a facility for importing and exporting standard keybind sets. This allows a site to maintain a library of keybind sets for the terminal types commonly used at the site. As delivered, uni-SPF includes a library of keybind sets for the following terminal types:

- aixterm — for the RS/6000 HFT keyboard
- sun-cmd — for the Sun-3 and Sun-4 (SPARCSta-tion) Type 3, 4, and 5 keyboard using the OpenWindows command tool
- hpterm — for the HP/9000 X-window console keyboard

Your site may add other keybind sets to this library. At program initiation, uni-SPF checks your current keybind table for entries that match your current TERM environment variable. If none are found, it then checks the library for a keybind set that matches your TERM environment variable. If one exists, it is automatically imported. The Keybind Import/Export option allows you to import other keybind sets for other terminals that you may use.

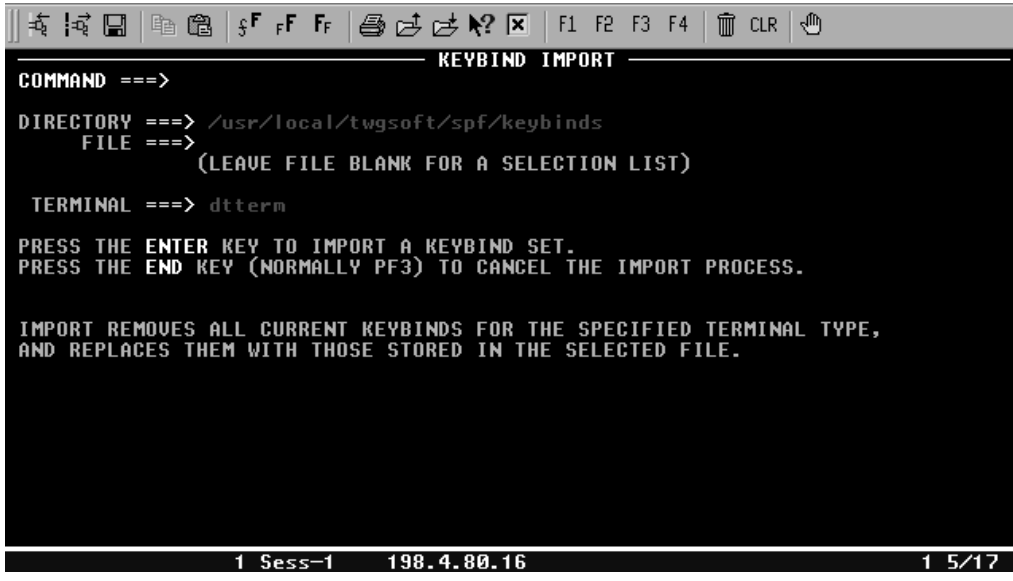
The Keybind Import/Export panel is shown on the following page.

Select 1 to import a keybind set. Select 2 to export a keybind set to the library.

The Keybind Import panel, Option 0.K.3.1 shown on the following page, allows you to specify the directory path and filename for the keybind set to be imported. The default directory path is the location of the keybind library delivered with the product. To access a different library, modify the directory path as needed.

The terminal type is automatically taken from the current setting of the TERM environment variable. To import keybinds for a different terminal type, change this field.

When you import a keybind set from the library, all existing keybinds for the terminal type specified are removed. The imported keybinds replace the previous definitions. If you wish to import a library keybind set without deleting your existing definitions, specify a different terminal type on the Keybind Import panel but



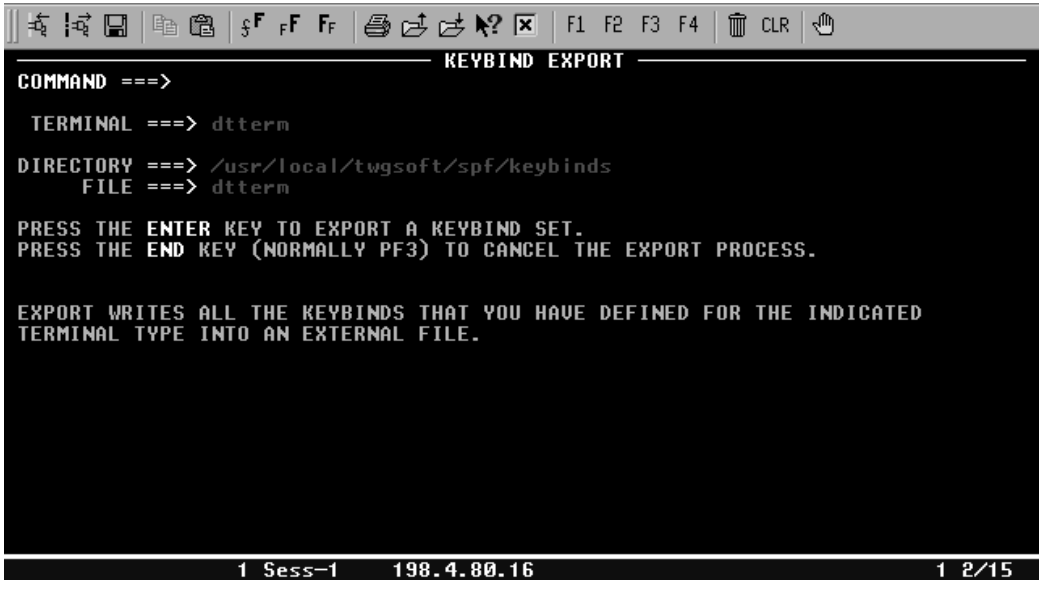
load the correct file name from the library. You may then use the Keybind Display/Update panel (Option 0.K.1) to selectively rename the terminal type for those keybinds that you wish to retain.

If you already know the name of the file containing the keybind set you wish to import, type its name in the file name field. To see a list of available files in the library, press Enter to display a File Selection panel.

On this panel, you may

- type “s” in the input field immediately to the left of one file to be imported. You may select only one file from this panel.
- use the SELECT command to choose one file to be imported. You may select only one file from this panel.
- use “s” or SELECT to change directories
- use scrolling commands or PF keys to reposition the display
- use the LOCATE command for a directed scroll

The Keybind Export panel, Option 0.K.3.2 shown below, allows you to specify the directory path and filename for the library where the new keybind set is to be stored. The default directory path is the location of the keybind library delivered with the product. To export



to a different library, modify the directory path as needed.

The terminal type is automatically taken from the current setting of the TERM environment variable. To export keybinds for a different terminal type, change this field.

Export stores **all** current settings for this terminal type in the library keybind set. To export only selected keybinds for this terminal type, use the Keybind Display/Update panel (Option 0.K.1) to assign temporary names to the keybinds that you do not wish to export.

The file name for the keybind set is also taken automatically from the current setting of the TERM environment variable. You may change this name to store the current keybind set in a different file.

Export requires write permission for the keybind library directory. Its use may therefore be restricted at some sites to the system administrator who has these privileges. If you attempt to export a keybind set and do not have the necessary privileges, you will receive one of the following messages:

- if the file for this keybind set does not exist:
  directory is not a directory to which you have write access

- if the file for this keybind set already exists:
  directory-path/*filename* is not a file to which you have write access

Chapter 4: Browse

Browse allows you to view the contents of a file but does not permit modifications. The display is similar to Edit, but the command set is limited to commands that allow you to view different sections of the file or to better understand what you are viewing. There are no line commands, and line numbers are not displayed.

When you choose Browse (Option 1) from the Main Menu panel, the Browse-Entry Panel appears.

The cursor is positioned initially at the file name input field. You may

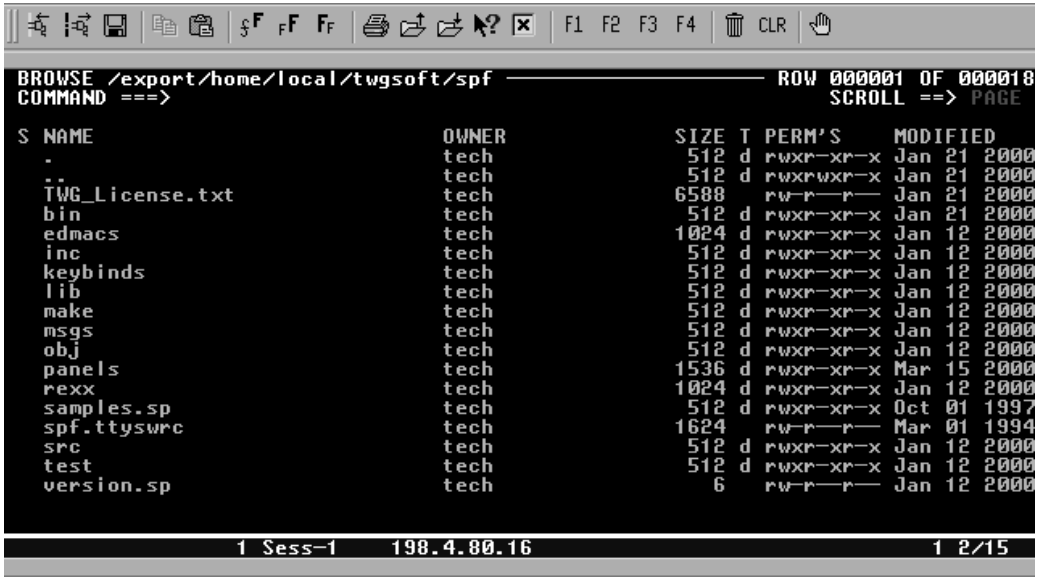
- complete the file name field with the name of a file to browse. This must be an existing file.
- reposition the cursor to the directory path field and type a valid directory path
- type both a valid directory path and the name of a file to browse
- leave the file name blank

If Browse is your first selection during your very first uni-SPF session, the directory path field is set to "." to indicate the current working directory. Subsequently, this field contains the name of the most recently used directory.

If you leave the file name blank, uni-SPF displays a file selection panel from which you select the file or files to be browsed. The file selection panel also shows the names of directories in the current path. By selecting a directory from this panel, you automatically modify the setting of the directory path field.

If you specify a filename that contains wild-card characters, the file selection panel displays only those files whose names match the template specified. ***Chapter 2: Interaction with uni-SPF*** contains a detailed discussion of using Unix wild-card characters when specifying filenames.

A sample Browse File Selection panel appears on the following page. Files and directories are listed in alphabetical order. Scrolling commands (or PF keys) are available to view portions of the list that cannot be displayed on a single screen. The LOCATE command is available to perform a directed scroll, repositioning a specific filename at the top of the display.



Select a file to be browsed by typing “s” in the input field immediately to the left of the filename. If you select a directory, uni-SPF changes the current directory path to the directory you selected and updates the file selection panel to reflect the contents of the new directory.

You may also use the SELECT command in the primary input field to select one or more files or to change the current directory. If you use the SELECT command, you **must** select files that already exist.

When you make the desired selections and press Enter, the browse display panel appears with the file you selected. The primary input field is the COMMAND line. The only other input field is the SCROLL field.

Browse commands are described in detail in this chapter. Scrolling and the setting of the SCROLL indicator is discussed in ***Chapter 2: Interaction with uni-SPF***.

Browse Ring

The uni-SPF browse facility includes a feature that is not standard in IBM's ISPF Program Product. The browse ring allows you to browse multiple files simultaneously within a single browse session.

You may start a browse session with one file or with multiple files. Regardless of how you initiate the session, you may add files to the ring or remove files from the ring at any time.

To start a browse session with multiple files select several files from the Browse File Selection panel before you press Enter. The files appear in the ring in the order selected. A message tells you how many files are in the ring and how to access the other files.

You may add a file to the ring at any time by using the BROWSE command.

```
BROWSE filename
```

adds the specified file to the ring and displays it as the current file to browse.

To view the next file in the ring, use BROWSE without any operands. BROWSE moves in a circular fashion through all files currently in the ring.

To view a specific file in the ring, use

```
BROWSE filename
```

This locates filename in the ring and displays it as the current file to browse.

To remove a file from the ring, use the END command when that file is currently displayed. The displayed file is removed and the next file in the ring appears.

To leave browse without processing each file separately, use END ALL. Alternatively, you may leave browse by using the "=" command or the RETURN command.

BROWSE

The BROWSE command manages files in the browse ring.

BROWSE [*filename*]
BRO
BR
B

With no operands, BROWSE displays the next file in the browse ring.

If ***filename*** is specified, uni-SPF searches the browse ring for ***filename***. If it is found, it is displayed as the current file. If it is not found, ***filename*** is added to the browse ring and displayed as the current file. You may specify a simple filename or a complete path. ***filename*** must already exist.

The section "Browse Ring" in this chapter discusses simultaneous browsing of multiple files in the browse ring.

COLUMNS

The COLUMNS command inserts a column-position line in the display.

COLUMNS
COLS
COL

The column-position line is inserted at the top of the display. It does not scroll with the data.

Use the RESET command to remove the column-position line.

DOWN

The DOWN command moves the display toward the bottom of the file. It is normally associated with the PF8 key.

DOWN **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, DOWN moves toward the bottom of the file by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the DOWN command to override the SCROLL field value.

You may use these operands in conjunction with a PF key by typing the operand on the command line and pressing the PF key associated with DOWN (normally, PF8).

Examples:

`DOWN`	scrolls toward the bottom of the file by the amount indicated in the SCROLL field
`DOWN 12`	scrolls down 12 lines
`DOWN M`	scrolls to the bottom of the file

END

The END command terminates browsing of the current file. If the current file is the last or the only file in the ring, END also terminates the current browse session and returns you to the previous panel. It is normally associated with the PF3 key.

END [ALL]

The ALL operand of END allows you to exit the current session without processing each file in the ring separately.

FIND

The FIND command searches for a string.

FIND ***s1*** **[NEXT]** **[CHARS]** **[*c1* [*c2*]]**
F **[ALL]** **[PREFIX]**
[FIRST] **[SUFFIX]**
[LAST] **[WORD]**
[PREV]

You may specify the command operands in any order. When you specify only the required string operand, FIND searches for ***s1*** within the entire width of the file. It searches for the next occurrence of ***s1***, beginning at the current cursor position. It treats ***s1*** as a character string (CHARS), meaning that it will find the string "CAN" as a single word or as a substring of other words or data as in "can", "candy", "pelican", or "decanter". It searches the entire file to locate ***s1***. The sections that follow describe how to specify strings and how to override the default actions of the FIND command.

If you use the ALL operand to find multiple occurrences of a string, the cursor is positioned on the first occurrence of the string. A message tells you how many occurrences of the string were found.

String Operands

s1 is the search string. This operand is required.

The string operands may be any of the following types of strings:

- simple strings
- delimited strings
- character strings
- hexadecimal strings
- picture strings
- asterisk strings

Simple strings are any combination of alphanumeric characters that is not enclosed in quotes. They may not contain blanks, commas, or asterisks. A simple string is not case sensitive. An example of a FIND command using simple strings is

```
FIND ABC
```

This command finds the next occurrence of “abc”, regardless of the case in which “abc” appears in the data.

Delimited strings are similar to simple strings but may contain blanks, commas, quotes, or asterisks. To insure that the string is processed properly, it must be enclosed in single or double quotes. If the string contains one form of quotes, it must be delimited by the other form. Examples of FIND commands that use delimited strings are

```
FIND ‘Once upon a time’
FIND “It’s grand”
```

Character strings provide case sensitivity. A character string is enclosed in quotes and immediately preceded or followed by the letter “C”. Examples of FIND commands with character strings are

```
F C’Have a nice day’
F ‘So long’C
```

Hexadecimal strings allow you to manipulate data which may not be represented by normal printing characters. A hexadecimal string is enclosed in quotes and immediately preceded or followed by the letter “X”. Because each byte of data is represented by two hexadecimal digits, hexadecimal values must be entered in pairs. The examples that follow find a tab character and a line feed character, respectively.

```
F ‘09’X
F X’0A’
```

Picture strings allow you to specify a template for the search string which may be matched by many different strings in the data. A picture string is enclosed in quotes and immediately preceded or followed by the letter "P". Picture strings may contain a combination of normal text characters and symbolic characters. The symbolic characters used to define the template are

=	any character
^	any non-blank character
.	any non-ASCII character (except the tab character, hexadecimal 09, which is treated as a displayable character)
#	any numeric character
-	any non-numeric character
@@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character (not numeric or alphabetic)

Examples of picture strings are

`P'N=='`	the letter N followed by any two characters; matches now, new, net, n43, n&y
`P'N@@'`	the letter N followed by any two alphabetic characters; matches now, new, and net but not n43 or n&y
`P'@##^=<'`	an alphabetic character followed by two numbers, any non-blank character, any character, and any lowercase character; matches "a23aAb",

"a23A b", and "Z99xxx" but not "a23 Ab" or "Z99xxX".

Asterisk strings allow you to reference strings previously used in a FIND command. If you specify ***s1*** as a single asterisk, that operand takes its value from the corresponding operand in the most recent FIND command issued.

Directional Operands

Directional operands control the direction of the search for ***s1***. If no directional operand is specified, the search begins at the current cursor position and moves forward. This is identical to specifying the NEXT operand. As alternatives, you may specify

PREV	the first occurrence of ***s1*** prior to the current cursor position
FIRST	the first occurrence of ***s1***, starting at the top of the file and searching forward
LAST	the first occurrence of ***s1***, starting at the bottom of the file and searching backward
ALL	all occurrences of ***s1***

String Location Operands

These operands allow you to define how ***s1*** is to be interpreted. The default is CHARS, where the string may be a single word or a substring of a word or other data.

Alternatives use the position of the string within the text to determine if the search is successful.

The position within the text is defined by the presence of a special character or a blank at the beginning or end of the string. For this purpose, a special character is any non-alphanumeric character except #, $, ‘, and @.

The alternative string location operands are

PREFIX The string must be preceded by a blank or special character. It may not be followed by a blank or special character.

SUFFIX The string must be followed by a blank or special character. It may not be preceded by a blank or special character.

WORD The string must be preceded and followed by a blank or special character.

The following sample line from a file is used to illustrate the string location operands. The string found is shown in bold.

PREFIX CAN ***CAN***DY PELICAN DECANTER ‘CAN’
*****CAN***NOT (AFRICAN)

SUFFIX CAN CANDY PELI***CAN*** DECANTER ‘CAN’
*CANNOT (AFR***I*****CAN***)

WORD ***CAN*** CANDY PELICAN DECANTER ‘***CAN***‘
*CANNOT (AFRICAN)

Column Operands

These operands allow you to restrict the search for ***s1*** to specific columns of data. If only ***c1*** is specified, the search is restricted to columns between ***c1*** and the right-most column of the file.

HEX

The HEX command is used to enable or disable hexadecimal display of data in the file being edited.

HEX [ON]
OFF

HEX with no operands is identical to HEX ON. Only the vertical display mode is supported. In this mode, three lines appear on the screen for each data line. The first is the normal character representation of the data. The next two lines are a vertical representation of the hexadecimal value corresponding to each character.

HEX OFF disables hexadecimal mode, returning the display to character representation only.

When you set HEX ON in conjunction with VIEW EXPAND, uni-SPF automatically changes the display to VIEW COMPRESS. This is necessary to ensure that the character and hexadecimal values align correctly in the display. If you subsequently set HEX OFF, the display returns to VIEW EXPAND.

LEFT

The LEFT command moves the display toward the left-most column of data. It is normally associated with the PF10 key.

LEFT **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, LEFT moves left by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the LEFT command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with LEFT (normally, PF10).

Examples:

`LEFT`	scrolls left in the file by the amount indicated in the SCROLL field
`LEFT 12`	scrolls left 12 columns
`LEFT M`	repositions the display so that the leftmost column appears in column 1

LOCATE

The LOCATE command performs a directed scroll of the data display.

LOCATE ***line-number***
LOC
L

LOCATE repositions the display so that the line specified by ***line-number*** is the top line of data. ***line-number*** is relative to the beginning of the file.

RESET

The RESET command removes the columns line inserted by the COLUMNS command.

RESET
RES

Since its function is restricted, the command has no operands.

RFIND

The RFIND command repeats the last FIND command. It is normally used from a PF key and is normally assigned to PF5.

RFIND

The RFIND command uses the operands specified on the most recently issued FIND command.

RIGHT

The RIGHT command moves the display toward the rightmost column of data. It is normally associated with the PF11 key.

RIGHT **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, RIGHT moves right by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the RIGHT command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with RIGHT (normally, PF11).

Examples:

`RIGHT`	scrolls right in the file by the amount indicated in the SCROLL field
`RIGHT 12`	scrolls right 12 columns
`RIGHT P`	scrolls right by the number of columns that normally appear in the current browse window

SHELL

The SHELL command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The SHELL command is synonymous with the ! command.

SHELL [*string*]

With no operands, SHELL takes you to the Unix shell. The specific shell used is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press Enter when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. If the Unix command displays information on the screen, the message

```
Press ENTER to continue uni-SPF
```

appears after the output of the Unix command.

UP

The UP command moves the display toward the top of the file. It is normally associated with the PF7 key.

UP **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, UP moves up by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the UP command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with UP (normally, PF7).

Examples:

`UP`	scrolls toward the top of the file by the amount indicated in the SCROLL field
`UP 12`	scrolls up 12 lines
`UP M`	scrolls to the top of the file

VIEW

The VIEW command controls the appearance on the screen of data that contains tab characters.

VIEW EXPAND
COMPRESS

With VIEW EXPAND, a tab character in the data automatically shifts the data to the column immediately following the next tab stop. This is the default. Tab stops are taken from the associated edit profile.

With VIEW COMPRESS, a tab character occupies a single character position in the data and no expansion occurs.

! (Exclamation Point)

The ! command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The ! command is synonymous with the SHELL command.

![*string*]

With no operands, ! takes you to the Unix shell. The specific shell used is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press Enter when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. If the Unix command displays information on the screen, the message

```
Press ENTER to continue uni-SPF
```

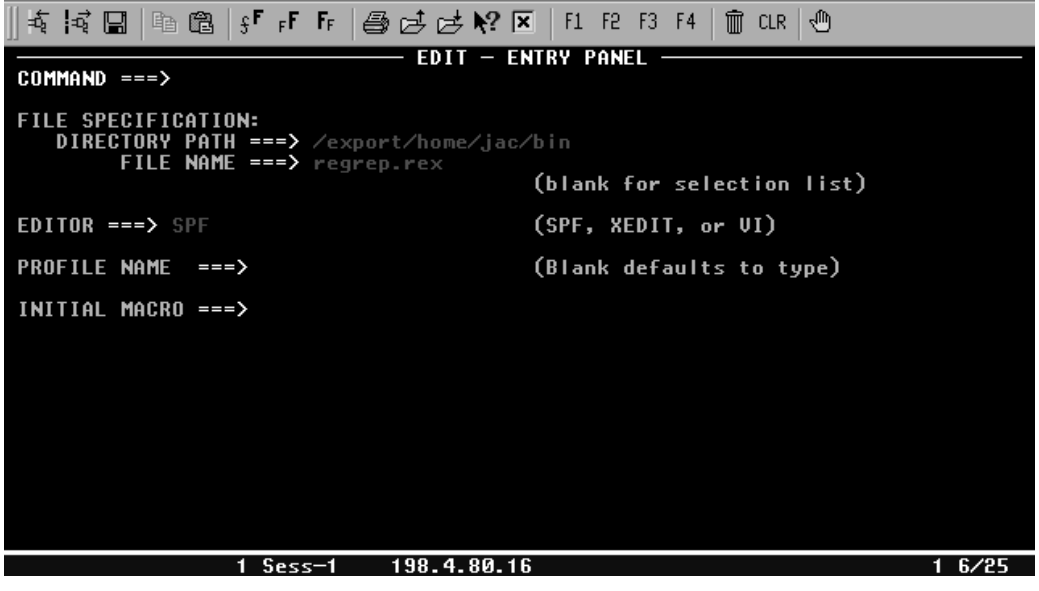
appears after the output of the Unix command.

Chapter 5: Edit

The standard uni-SPF editor is modelled on the PDF editor component of IBM's ISPF Program Product. To distinguish it from available editor alternatives, it is referred to as the SPF editor.

The uni-SPF editor also supports an "edit ring" that allows you to edit several files simultaneously. The EDIT command allows you to switch between files in the ring or to add new files to the edit ring. This feature is discussed further in the section entitled "Edit Ring".

When you choose Edit (Option 2) from the Main Menu panel, the Edit-Entry Panel appears.



The cursor is positioned initially at the file name input field. You may

- complete the file name field with the name of a file to edit. This may be an existing file or a new file to be created.
- reposition the cursor to the directory path field and type a valid directory path
- type both a valid directory path and the name of a file to edit
- leave the file name blank
- optionally, select an alternative editor. This manual describes the use of the default uni-SPF editor, which is an SPF editor. Available alternatives are uni-XEDIT and vi. If you select either of these editors, refer to their respective user manuals for information on editing commands and features.
- optionally, specify a profile to use for this edit session
- optionally, specify an initial macro to run before each file in the ring is displayed

If Edit is your first selection during your very first uni-SPF session, the directory path field is set to "." to indicate the current working directory. Subsequently, this field contains the name of the most recently used directory.

Use the Profile Name field to specify the profile for all files in this edit session. This profile is used regardless of whether you specify the file name on the Edit Entry panel or select one or more files from a file selection list. It is also used for files added to the ring during the session using the EDIT primary command. This value is saved across uni-SPF sessions in your uni-SPF profile (SPFPROF.tbl). To edit a file with a different profile, use SPLIT (normally assigned to PF2) to split the screen and start a new uni-SPF session.

Use the Initial Macro field to specify a macro to be run before the file is displayed for all files in this edit session. This macro is run regardless of whether you spec-

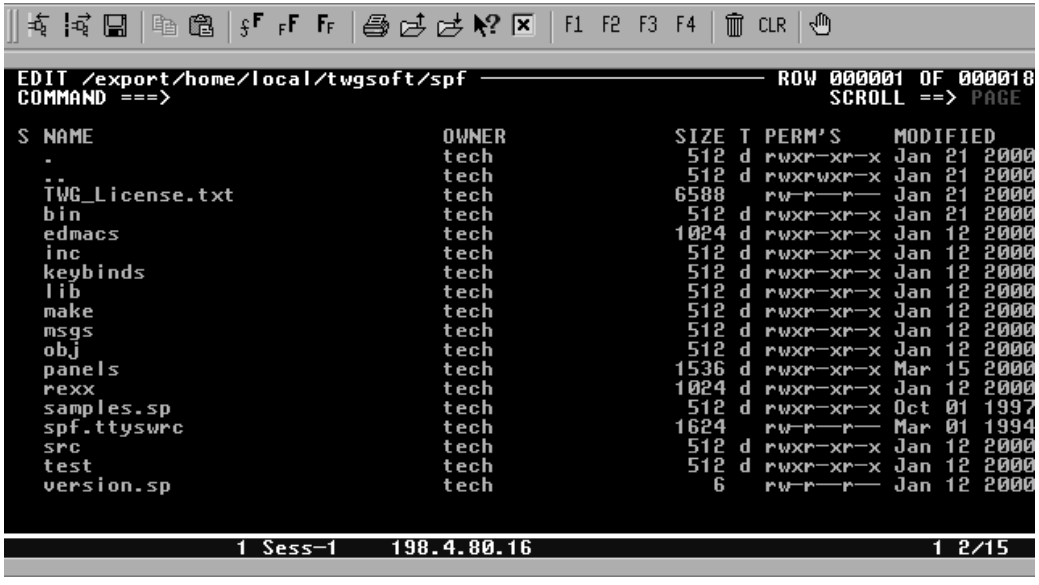
ify the file name on the Edit-Entry panel or select one or more files from a file selection list. It is also run for files added to the ring during the session using the EDIT primary command. This value is **not** saved in your uni-SPF profile. To edit a file with a different or no initial macro, use SPLIT (normally assigned to PF2) to split the screen and start a new uni-SPF session.

If an initial macro is specified on the Edit-Entry Panel, it takes precedence over any IMACRO setting in the edit profile.

If you leave the file name blank, uni-SPF displays a file selection panel from which you select the file or files to be edited. The file selection panel also shows the names of directories in the current path. By selecting a directory from this panel, you automatically modify the setting of the directory path field.

If you specify a filename that contains wild-card characters, the file selection panel displays only those files whose names match the template specified. ***Chapter 2: Interaction with uni-SPF*** contains a detailed discussion of using Unix wild-card characters when specifying filenames.

A sample Edit File Selection panel appears on the following page. Files and directories are listed in alphabetical order. Scrolling commands (or PF keys) are available to view portions of the list that cannot be displayed on a single screen. The LOCATE command is available to perform a directed scroll, repositioning a specific filename at the top of the display.



Select a file to be edited by typing “s” in the input field immediately to the left of the filename. If you select a directory, uni-SPF changes the current directory path to the directory you selected and updates the file selection panel to reflect the contents of the new directory. For simultaneous editing of multiple files, select several files to be edited before you press Enter.

You may also use the SELECT command in the primary input field to select one or more files or to change the current directory. The SELECT command has syntax and performance specific to the Edit File Selection panel which is documented in the next section.

SELECT

The SELECT command allows you to select one or more files from a file selection panel or to change the current directory in a file selection panel.

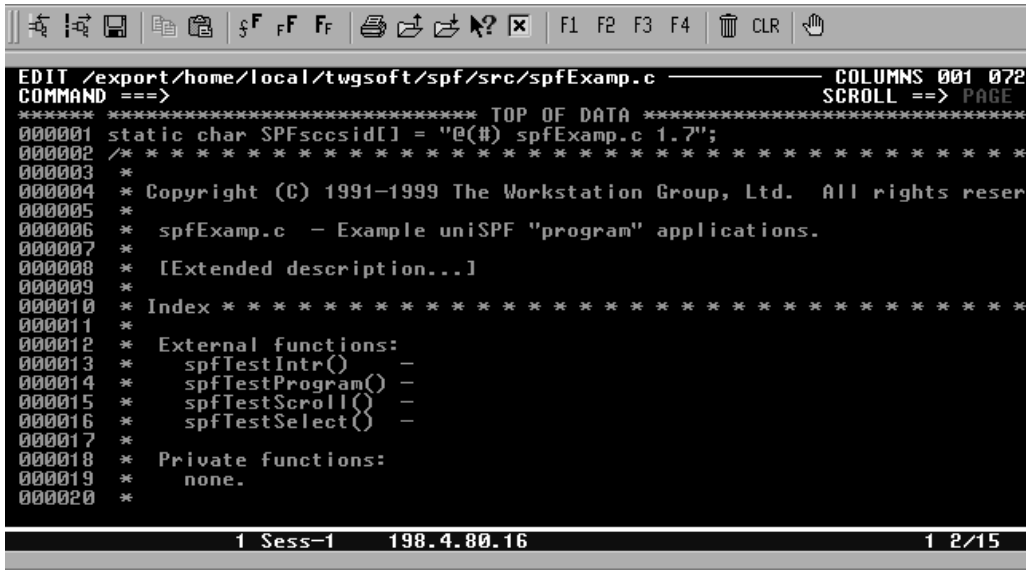
SELECT ***name-1*** **[. . .** ***name-n*****]**
SEL
S

If any ***name*** specified is a directory, uni-SPF changes the current directory to the first one specified and ignores all other ***name***s on the SELECT command.

If all ***name***s specified are files, SELECT adds all of the files to the edit ring in the order specified on the command.

You may use the SELECT command in conjunction with the "s" command in the input area to the left of the filenames. If you select only files, the files specified on the SELECT command are placed in the ring ahead of those marked with an "s". If one or more of the entries marked with an "s" is a directory and the SELECT command specified only filenames, uni-SPF changes the current directory to the first one in the file selection list that is marked with an "s". If the SELECT command specified a directory name, uni-SPF changes the current directory to the first one specified and ignores all other SELECT command operands and all entries marked with an "s".

When you make the desired selections and press Enter, the edit display panel appears with the file or files you selected for editing.



The edit display panel has the following input fields:

- **COMMAND line**, where edit Primary Commands are entered
- **SCROLL indicator**, where the default scrolling value is set
- **Line command area**, where edit Line Commands are entered
- **Data area**, where data from the file is displayed and modified

Edit Primary Commands are described in detail in this chapter. Scrolling and the setting of the SCROLL indicator is discussed in ***Chapter 2: Interaction with uni-SPF.***

Edit Line Commands are described in detail in this chapter. The line command area normally contains sequential line numbers. These line numbers are not saved with the data. When you create a new file or add lines to an existing file, the line command area of

such new data lines contains apostrophes. When you place data in a new line and press Enter, the apostrophes are changed to appropriate line numbers.

Contents of the file are displayed in the data area. The data may be changed by using an editor command or by positioning the cursor in the proper location and typing directly in the data area.

Edit Ring

The uni-SPF editor includes a feature that is not standard in IBM's ISPF Program Product. The edit ring allows you to edit multiple files simultaneously within a single edit session.

You may start an edit session with one file or with multiple files. Regardless of how you initiate the session, you may add files to the ring or remove files from the ring at any time.

To start an edit session with multiple files select several files from the Edit File Selection panel before you press Enter. The files appear in the ring in the order selected. MSG lines tell you how many files are in the ring and how to access the other files.

You may add a file to the ring at any time by using the EDIT command.

```
EDIT filename
```

adds the specified file to the ring and displays it as the current file being edited.

To view the next file in the ring, use EDIT without any operands. EDIT moves in a circular fashion through all files currently in the ring.

To view a specific file in the ring, use EDIT ***filename***. This locates ***filename*** in the ring and displays it as the current file being edited.

To remove a file from the ring, use the appropriate CANCEL, END, or SAVE command when that file is

currently displayed. The displayed file is removed and the next file in the ring appears.

To leave the editor without processing each file separately, use either END ALL or CANCEL ALL. END ALL uses the AUTOSAVE setting for each file to determine its disposition. CANCEL ALL cancels all changes that may have been made to any file in the ring.

Alternatively, you may leave the editor by using the "=" command or the RETURN command. Both "=" and RETURN use the AUTOSAVE setting to determine the disposition of each file. If any file in the ring has AUTOSAVE set to PROMPT and changes have been made to that file, the "=" or RETURN command is interrupted by the editor prompt to SAVE or CANCEL changes to the file. You must then re-enter the "=" or RETURN command to proceed.

Edit Profile

Each time you edit a file, uni-SPF initiates a specific editing environment for that file. This environment consists of certain option and mode settings that can be modified by editor Primary Commands. This environment is known as the edit profile. Non-default settings of these options or modes are stored in a profile table (ISPEPROF.tbl) in each user's HOME directory

The editor profile settings available in the current release of uni-SPF are

AUTOSAVE	Determines whether data is automatically saved when you end a session
BOUNDS	Column boundaries for various commands
CAPS	Controls case of input data
HEX	ASCII or ASCII with hexadecimal display
IMACRO	Specifies if an initial macro is to be run before displaying the file
OVERWRITE	Determines if SAVE operations write directly to the original file
PROFILE	Controls updating of profile table
TABS	Controls behavior of tab and backtab keys

If the file does not use one of the language-specific profiles discussed below, the default settings for each of these modes and options are

AUTOSAVE	ON
BOUNDS	1, 4096
CAPS	OFF
HEX	OFF
IMACRO	NONE
OVERWRITE	OFF
PROFILE	UNLOCK
TABS	ON

Refer to the description of the primary commands for each of these options to determine the possible settings and their effects on the data. The PROFILE primary command describes how to display the current settings and how to invoke a different profile.

The profile name is normally taken from the final component of the filename. In the case of a simple file-name, such as “mydata”, the profile name is identical to the filename. In the case of a more complex filename, such as “my.new.data”, the profile name is “data”.

Some programming languages have conventions for naming files that include a specific extension to identify the type of source file. As an example, source code written in the C language is typically named something like “srcprog.c”. For such files, the editor profile name is “c”.

uni-SPF includes standard editor profiles for each of the following programming languages:

Language	File Extension	Profile Name
C	.c	c
C Intermediate(*)	.i	i
C headers	.h	h
C++	.cc	cc
C++ headers	.hh	hh
Assembler	.s	s
Fortran	.f	f
Pascal	.pas	pas

(*) after pre-processing

The CAPS, BOUNDS, and TABS settings for each of these profiles is shown in the following table:

Profile Name	BOUNDS	CAPS	TABS
c	1, 4096	OFF	every 8 columns
i	1, 4096	OFF	every 8 columns
h	1, 4096	OFF	every 8 columns
cc	1, 4096	OFF	every 8 columns
hh	1, 4096	OFF	every 8 columns
s	1, 4096	OFF	every 8 columns
f	1, 72	ON	1, 7, 40
pas	1, 4096	OFF	every 8 columns

Option 0.7 is also available to define or modify edit profiles.

Protection Against Loss of Data

Multiple Edit Protection. The action of the SAVE command and of automatic saving when AUTOSAVE is ON provides some protection for multiple users editing the same file. In this context, multiple users may be any of the following:

- two or more userids editing the same file with uni-SPF
- two or more userids editing the same file with uni-SPF and possibly other editors
- one userid editing the same file in two or more windows with uni-SPF and possibly other editors

Because of the way this protection is implemented, it is effective across all NFS links as well as on the current workstation.

When you bring a file into the editor, uni-SPF captures the “last modified” date/time stamp of the file. When you later enter a SAVE command or end an edit session

with AUTOSAVE ON, uni-SPF again checks the "last modified" date/time stamp of the disk file before performing the save. If the current date/time stamp does not match the original date/time, uni-SPF displays a message.

You then have the option to save the file under another name using CREATE. This action protects any changes that may have been made by another user. REPLACE bypasses the date/time stamp checking and overwrites the existing file. If you use REPLACE with the original filename, this action destroys any changes that may have been made by another user.

File-System-Full Protection. uni-SPF also provides protection against loss of data in the event of file-system-full conditions. Two alternative protection mechanisms are available as well as options to disable this protection.

When saving a file, uni-SPF protects data by copying the current edit session to a temporary file, removing the original file, and replacing it with the temporary copy. In the event of a file-system-full condition, a message appears indicating the condition and the location of the preserved copy. In the event that the temporary file cannot be created, you may use REPLACE to bypass this protection mechanism.

The environment variable SPFTEMP may be used to control the location of the temporary file or to bypass the protection mechanism entirely. The OVERWRITE command also allows you to disable the protection mechanism. SPFTEMP applies to the entire uni-SPF session for which it is set. The OVERWRITE setting is stored in the file's edit profile and may therefore be set uniquely for each file.

When SPFTEMP is not defined, uni-SPF writes the temporary file in the same directory as the original file. This is the default behavior.

When SPFTEMP is set to a fully qualified directory name, uni-SPF writes the temporary file to this directory.

When SPFTEMP is set to "NOTEMP", uni-SPF disables file protection completely. No temporary file is created and all attempts to save the file write directly to the original file. (NOTEMP must be specified in uppercase and may not be abbreviated.)

When OVERWRITE is set to ON, uni-SPF disables file protection completely. No temporary file is created and all attempts to save the file write directly to the original file.

The most complete protection is provided when SPFTEMP identifies a directory with ample free space in a file system other than the one where user files typically reside. This directory must have write permission for all users and ample space to save temporary copies of your largest files.

Excellent protection is also provided through the default mechanism, when SPFTEMP is not set. If a file-system-full condition prevents creation of the temporary file, you may save the file to a different file system by specifying a full path name on the CREATE command. Alternatively, you may use REPLACE to bypass the protection mechanism.

The greatest exposure occurs when SPFTEMP is set to "NOTEMP" or OVERWRITE is set ON. This disables protection completely and leaves you vulnerable to loss of data. Any attempt to save a file writes directly over the original file. If a file-system-full condition occurs, the output file is truncated at that point and the updated file exists **only** in the current session.

Primary Commands

Editor Primary Commands operate on the entire file or on a specified range of data in the file. The syntax of a primary command is generally in the form of a command followed by required and/or optional operands. Primary Commands are typed on the COMMAND line, which is the primary input field of the edit display panel.

The edit Primary Commands are

AUTOSAVE
BOUNDS
BUILTIN
CANCEL
CAPS
CHANGE
COMPRESS
COPY
CREATE
CUT
DEFINE
DELETE
DOWN
EDIT
END
ENDSAVE
EXCLUDE
EXPAND
FIND
HEX
IMACRO
HOME
LEFT
LOCATE
MOVE
OVERWRITE
PASTE
PROFILE
RCHANGE
REPLACE
RESET
RFIND
RIGHT
SAVE
SHELL
SORT
SUBMIT
TABDEF
TABS
UP
VIEW
!

The pages which follow contain a detailed description of each of the primary commands. A discussion of line commands and descriptions of each line command follow the primary commands.

AUTOSAVE

The AUTOSAVE command sets the autosave mode to control saving of the file when you end the edit session. It is identical to the ENDSAVE primary command.

AUTOSAVE [ON]
[OFF [PROMPT]]
[NOPROMPT]

Autosave mode has an effect only when data in the file has been changed. Data is considered to be changed when you actually modify the data, even if you later change it back to its original state. Data is also considered to be changed when you execute a command that has the potential for change, even if no change results (such as shifting a blank line or changing a string to itself). You may specify the command operands in any order.

AUTOSAVE with no operands is identical to AUTO-SAVE ON. When autosave mode is on, the file is automatically saved when you end the edit session using the END primary command (or PF3).

AUTOSAVE OFF is identical to AUTOSAVE OFF PROMPT. In autosave prompt mode, a prompt appears when you end the session. You may then enter SAVE to save your changes or CANCEL to terminate the session without saving.

AUTOSAVE OFF NOPROMPT turns autosave mode off and suppresses prompting. In this case, END behaves exactly like CANCEL. To save changes, you **must** enter a SAVE command before you press PF3.

BOUNDS

The BOUNDS command sets the left and right column boundaries and saves this information in the edit pro-file. It has the same effect as setting the boundaries on the =BNDS> line that is inserted by the BOUNDS line command. When the BOUNDS primary command is used in conjunction with the line command, the line command takes precedence.

BOUNDS [***left-column right-column***]
BOUND
BNDS
BND

BOUNDS with no operands resets the column boundaries to the default columns, 1 and 4096.

If specified, ***left-column*** identifies the left column boundary and ***right-column*** identifies the right column boundary. You may use * for either boundary to specify the current setting for that boundary. You may not specify the same column for both boundaries.

Column boundaries limit the scope of the following editor operations:

- left-shift and right-shift line commands
- left and right scrolling
- FIND, CHANGE, and EXCLUDE primary commands when you do not specify explicit columns as part of the command
- O, OO line commands
- TS line command

The effect of BOUNDS settings on scrolling is special. Scrolling to the left or right stops at the left or right bound, respectively. A subsequent scroll in the same direction scrolls beyond the bound, assuming that the bound is not the leftmost or rightmost column position.

Examples:

`BNDS 1 72`	sets the left and right column boundaries at 1 and 72, respectively
`BOUNDS`	resets the column boundaries to the defaults (1 and 4096)
`BND 9 *`	sets the left column boundary to 9 and retains the current setting for the right boundary

BUILTIN

Use the BUILTIN command to execute an editor command whenever a macro of the same name has been defined.

BUILTIN *command*

command may be any editor primary command.

As an example, you may choose to associate your macro MYDOWN with the name DOWN by issuing the command DEFINE MYDOWN ALIAS DOWN. When you type DOWN, the editor automatically executes the macro MYDOWN. To execute uni-SPF's DOWN command, you must type BUILTIN DOWN.

CANCEL

The CANCEL command terminates edit of the current file without saving any of your changes. If the current file is the last or the only file in the ring, CANCEL (without operands) also ends the current edit session. It allows you to discard changes made in the current session regardless of the autosave mode setting.

CANcel [ALL]

If you use the SAVE command during the current session, only those changes made since the last SAVE are discarded when you enter CANCEL. All changes prior to the last SAVE are preserved.

The ALL operand allows you to exit the current session without processing each file in the ring separately.

CAPS

The CAPS command controls the setting of caps mode. Caps mode affects the case of data entered at the terminal and of strings that are operands of the FIND and CHANGE commands. It does not affect operands of other primary commands entered on the command line. Specifically, filenames entered as a command operand are not affected by the CAPS setting.

CAPS [ON]
OFF

CAPS without an operand is identical to CAPS ON. When caps mode is ON, alphabetic data entered at the terminal is automatically translated to upper case. String operands of FIND and CHANGE are also automatically translated to upper case.

When caps mode is OFF, the case of data and of string arguments remains as you type it.

CHANGE

The CHANGE command searches for a string and changes it to a different string.

CHANGE ***s1*****s2**[***r***][**NEXT**][**CHARS**] [**X**][***c1***[***c2***]]
CHG [**ALL**] [**PREFIX**] [**NX**]
C [**FIRST**] [**SUFFIX**]
[**LAST**] [**WORD**]
[**PREV**]

You may specify the command operands in any order. When you specify only the required string operands, CHANGE searches for ***s1*** within the column boundaries specified by the current BOUNDS setting. It searches for the next occurrence of ***s1***, beginning at the current cursor position. It treats ***s1*** as a character string (CHARS), meaning that it will find the string "CAN" as a single word or as a substring of other words or data as in "can", "candy", "pelican", or "decanter". It searches the entire file to locate ***s1***. The sections that follow describe how to specify strings and how to override the default actions of the CHANGE command.

If you use the ALL operand to change multiple occurrences of a string, each changed line is identified by the appearance of ==CHG> in the line command area. If ***s2*** is longer than ***s1*** with the result that the replacement would exceed the current right bound, each such line is identified by the appearance of ==ERR> in the line command area. No changes are made. Use the RESET command to remove these indicators.

String Operands

s1 is the search string. ***s2*** is the replacement string. These two operands are required.

The string operands may be any of the following types of strings:

- simple strings
- delimited strings
- character strings
- hexadecimal strings
- picture strings
- asterisk strings

Simple strings are any combination of alphanumeric characters that is not enclosed in quotes. They may not contain blanks, commas, or asterisks. A simple string is not case sensitive. An example of a CHANGE command using simple strings is

```
CHANGE ABC DEF
```

This command changes the next occurrence of “abc” to “def”, regardless of the case in which “abc” appears in the data. The case of the replacement string in the data is determined by the CAPS setting.

Delimited strings are similar to simple strings but may contain blanks, commas, quotes, or asterisks. To insure that the string is processed properly, it must be enclosed in single or double quotes. If the string contains one form of quotes, it must be delimited by the other form. Use delimited strings to change existing data to a null string. Examples of CHANGE commands that use delimited strings are

```
CHG ‘Once upon a time’ ‘Long, long ago’
CHG “It’s grand” “That’s nice”
CHG very ‘’
```

Character strings provide case sensitivity. A character string is enclosed in quotes and immediately preceded or followed by the letter “C”. Examples of CHANGE commands with character strings are

```
C C’Have a nice day’ C’Bye-bye’
C farewell ‘So long’C
```

Hexadecimal strings allow you to manipulate data which may not be represented by normal printing characters. A hexadecimal string is enclosed in quotes and immediately preceded or followed by the letter "X". Because each byte ofdata is represented by two hexadecimal digits, hexadecimal values must be entered in pairs. The examples that follow change a tab character to one or more blanks or null:

```
C '09'X '20'X
C X'09' ''
C X'09' '2020'X
```

Picture strings allow you to specify a template for the search string (***s1***) which may be matched by many different strings in the data. A picture string is enclosed in quotes and immediately preceded or followed by the letter "P". Picture strings may contain a combination of normal text characters and symbolic characters. The symbolic characters used to define the template are

=	any character
^	any non-blank character
.	any non-ASCII character (except the tab character, hexadecimal 09, which is treated as a displayable character)
#	any numeric character
-	any non-numeric character
@@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character (not numeric or alphabetic)

Examples of ***s1*** picture strings are

```
P'N=='
```

	the letter N followed by any two characters; matches now, new, net, n43, n&y
`P'N@@'`	the letter N followed by any two alphabetic characters; matches now, new, and net but not n43 or n&y
`P'@##^=<'`	an alphabetic character followed by two numbers, any non-blank character, any character, and any lowercase character; matches “a23aAb”, “a23A b”, and “Z99xxx” but not “a23 Ab” or “Z99xxX”.

For the CHANGE command, ***s1*** may be any valid picture string. However, ***s2*** must conform to these rules:

- ***s2*** must be identical in length to ***s1***
- the only valid symbolic characters in ***s2*** are
  - `=` the corresponding character in ***s1*** is unchanged
  - `<` if the corresponding character in ***s1*** is uppercase alphabetic, convert it to lowercase
  - `>` if the corresponding character in ***s1*** is lowercase alphabetic, convert it to uppercase

Asterisk strings allow you to reference strings previously used in a FIND, CHANGE, or EXCLUDE command. If you specify either ***s1*** or ***s2*** as a single asterisk, that operand takes its value from the corresponding operand in the most recent CHANGE command issued.

Range Operand

Use the range operand, ***r***, to limit the portion of the file to be searched for ***s1***. ***r*** must be specified as labels in the form “*.label1 .label2*”. If you specify user-defined

labels, the labels must have been defined prior to issuing the CHANGE command. You may also specify a system-defined label, such as .zfirst or .zlast.

Directional Operands

Directional operands control the direction of the search for ***s1***. If no directional operand is specified, the search begins at the current cursor position and moves forward. This is identical to specifying the NEXT operand. As alternatives, you may specify

PREV	the first occurrence of ***s1*** prior to the current cursor position
FIRST	the first occurrence of ***s1***, starting at the top of the file and searching forward
LAST	the first occurrence of ***s1***, starting at the bottom of the file and searching backward
ALL	all occurrences of ***s1***

String Location Operands

These operands allow you to define how ***s1*** is to be interpreted. The default is CHARS, where the string may be a single word or a substring of a word or other data. Alternatives use the position of the string within the text to determine if the search is successful.

The position within the text is defined by the presence of a special character or a blank at the beginning or end of the string. For this purpose, a special character is any non-alphanumeric character except #, $, ‘, and @.

The alternative string location operands are

PREFIX The string must be preceded by a blank or special character. It may not be followed by a blank or special character.

SUFFIX The string must be followed by a blank or special character. It may not be preceded by a blank or special character.

WORD The string must be preceded and followed by a blank or special character.

The following sample line from a file is used to illustrate the string location operands. The string found is shown in bold.

PREFIX CAN ***CAN***DY PELICAN DECANTER 'CAN'
*****CAN***NOT (AFRICAN)

SUFFIX CAN CANDY PELI***CAN*** DECANTER 'CAN'
*CANNOT (AFRI***CAN***)

WORD ***CAN*** CANDY PELICAN DECANTER '***CAN***'
*CANNOT (AFRICAN)

Scope Operands

These operands allow you to restrict the search for ***s1*** to excluded (X) or non-excluded (NX) lines.

Column Operands

These operands allow you to restrict the search for ***s1*** to specific columns of data. If only ***c1*** is specified, the search is restricted to columns between ***c1*** and the right boundary column from the current BOUNDS setting.

COMPRESS

The COMPRESS command allows you to convert blank spaces into tab characters.

COMPRESS

COMPRESS removes strings of blanks and replaces them with one or more tab characters. The current hardware tab settings are used to determine the number of tab characters required. Tab characters are indicated by the ">" character in the data. The resulting data looks identical to the original when you set VIEW EXPAND.

Use the EXPAND primary command to convert tab characters to blank spaces.

Refer to the VIEW command for information on controlling the display of data containing tab characters.

COPY

Use the COPY command to copy a file from disk into the file that is currently being edited. You may copy the entire file or selected lines.

COPY [*filename*] [BEFORE *label*]
[AFTER *label*]

You may specify the command operands in any order.

filename is the name of the file to be copied from disk. If you specify only a filename, uni-SPF assumes that the file is located in the current working directory. You may specify a full path name to identify the file to be copied. Omit this operand to display the Edit-Copy panel.

If the file you are editing is not empty, you must specify a destination for the lines that are copied. The simplest way to do this is to use the A (after) or B (before) line command. Alternatively, use the `BEFORE ***label***' or `AFTER ***label***' operand of the COPY command. If ***label*** is a user-defined label, it must have been defined prior to the copy. ***label*** may also be a system-defined label, such as .zfirst or .zlast. Note that the COPY command does not recognize the optional numeric specification of the A or B line command.

If you do not specify a destination, the message "COPY pending" appears, and your copy command remains on the command line. You may provide a destination using a line command, modify your command to add the appropriate operand, or erase the command to cancel its execution.

If you do not specify a ***filename***, the Edit-Copy panel appears. On this panel you may provide the directory path and file name to be copied. You may also specify a range of the file to copy by completing the First Line Number and Last Line Number fields on this panel. Line numbers are relative to the beginning of the file and have no relationship to any line numbers that may

be displayed or stored with the file. **This is the only way to limit the amount of data that is copied**. Otherwise the entire file is copied into the file currently being edited at the destination specified.

Examples:

`COPY FILE1`

copies the entire file named FILE1 from the current working directory into the file being edited. If the file being edited already contains data, you must use the A (after) or B (before) line command to indicate the destination for the copy.

`COPY FILE1 BEFORE .ZFIRST`

copies FILE1 in the current working directory into the file being edited, placing the data from FILE1 before the first data line in the file being edited.

`COPY AFTER .LABL2`

establishes the destination for a copy from a disk file. The Edit-Copy panel appears for you to identify the file that is the source for the copy. You may also specify the range of the file that you wish to copy.

`COPY BEFORE .ZLAST /home/subdir/myfile`

copies the file /home/subdir/myfile and places the copied lines immediately preceding the last data line in the file being edited.

CREATE

The CREATE command creates a new file from data in the file currently being edited. Either all or a portion of the current file may be written to the new file.

CREate [*filename*] [*range*]

You may specify the command operands in any order.

filename is the name of the file to be created. If you specify only a ***filename***, uni-SPF assumes that the file is to be placed in the current working directory. You may specify a full path name to identify the file to be created. Omit this operand to display the Edit-Create panel.

You must specify a range to be written to the new file, even if you want to use all the data in the file currently being edited. This is because CREATE differentiates between copying data to the new file and moving data from the current file to the new file. The easiest way to designate the range to be written is to use the appropriate C (copy) or M (move) line command. C, C***n***, ***n***C, or CC designate lines to be copied to the new file. M, M***n***, ***n***M, or MM designate lines to be moved to the new file and deleted from the current file.

Alternatively, you may specify the range as an operand of the CREATE command. ***range*** must be specified using labels in the form "*.label1 .label2*". If you specify user-defined labels, the labels must have been defined prior to issuing the CREATE command. You may also specify a system-defined label, such as .zfirst or .zlast. Using the range operand **copies** data to the new file.

If you specify the name of a file that already exists, CREATE does not overwrite it. Instead, the message "***filename*** already exists" appears, and the line commands that designate the range to be written remain pending. Use the REPLACE primary command to overwrite an existing file.

If you do not specify a ***filename***, the Edit-Create panel appears. On this panel you may provide the directory path and file name to be copied.

Examples:

`CREATE VERSION2`

creates a new file named VERSION2 in the current working directory. Copies lines marked in the current file with the C line command or moves lines marked in the current file with the M line command.

`CREATE .ZFIRST .ZLAST`

creates a new file in the current working directory. All lines of the current file are copied. The Edit-Create panel appears for you to identify the file to be created.

`CREATE /home/owner/subdir/version3 .START .STOP`

creates a new file named version3 in the directory /home/owner/subdir. Lines are copied from the current file beginning at the label .start and ending with the label .stop.

CUT

The CUT command copies one or more lines from the file to the cut buffer.

CUT

The line(s) to be cut must be identified by a C or M line command. All forms of the C and M line commands are valid for use with CUT.

The previous contents of the cut buffer are replaced by the results of the current CUT command. The contents of the cut buffer remain constant until you issue another CUT command or until you exit uni-SPF.

The cut buffer is stored in the user's HOME directory in a file named ".spf.clipboard". You must have write permission for this file.

Use the PASTE command to retrieve the contents of the cut buffer.

DEFINE

The DEFINE command establishes a synonym for an editor macro name.

DEFINE *synonym* ALIAS *macro-name*
DEF

macro-name is the name of the macro that is to be executed whenever you type ***synonym*** on the command line.

Examples:

```
DEFINE FETCH ALIAS GETFILES
```

establishes FETCH as a synonym for the macro GETFILES. When you type FETCH on the command line, the macro named GETFILES is executed.

DELETE

The DELETE command deletes one or more lines from the file you are currently editing. An operand allows you to restrict operation of the command to excluded or non-excluded lines.

DELete ALL [X] [*range*] [NX]

You may specify the command operands in any order. The operand ALL is required.

X restricts the deletion to excluded lines. NX restricts the deletion to non-excluded lines. If you specify either X or NX, it is not necessary to specify a range. If you specify a range with X or NX, excluded (or non-excluded) lines outside the range are not deleted.

If you omit X or NX, you must specify a range to define the lines to be deleted. ***range*** must be specified using labels in the form "*.label1 .label2*". If you specify user-defined labels, the labels must have been defined prior to issuing the DELETE command. You may also specify a system-defined label, such as .zfirst or .zlast.

To delete a single line, use the same label for the beginning and the end of the range. Alternatively, use the Delete line command.

Examples:

```
EXCLUDE ALL 'Comments:'
DELETE ALL X
```

deletes all lines in the file that contain the string `Comments:' by first excluding those lines and then deleting all excluded lines

```
DEL ALL NX .START .STOP
```

deletes all non-excluded lines between the labels .START and .STOP. Non-excluded lines outside this range are preserved.

`DEL .START .STOP ALL`

deletes all lines between the labels .START and .STOP.

`DEL ALL .PURGE .PURGE`

deletes the single line labelled .PURGE.

DOWN

The DOWN command moves the display toward the bottom of the file. It is normally associated with the PF8 key.

DOWN [Page]
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, DOWN moves toward the bottom of the file by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the DOWN command to override the SCROLL field value.

You may use these operands in conjunction with a PF key by typing the operand on the command line and pressing the PF key associated with DOWN (normally, PF8).

Examples:

`DOWN`	scrolls toward the bottom of the file by the amount indicated in the SCROLL field
`DOWN 12`	scrolls down 12 lines
`DOWN M`	scrolls to the bottom of the file

EDIT

The EDIT command manages files in the edit ring. The section "Edit Ring" in this chapter discusses simultaneous editing of multiple files in the edit ring.

EDIT [*filename*]
ED
E

With no operands, EDIT displays the next file in the edit ring. If ***filename*** is specified, uni-SPF searches the edit ring for ***filename***. If it is found, it is displayed as the current file. If it is not found, ***filename*** is added to the edit ring and displayed as the current file. You may specify a simple filename or a complete path. ***filename*** may be an existing file or a new file to be created.

END

The END command terminates edit of the current file. If the current file is the last or only file in the ring, END (with no operands) also terminates the current edit session and returns you to the previous panel. It is normally associated with the PF3 key.

END [ALL]

Whether or not changes to the file are saved depends on your setting of autosave mode. If autosave is ON, all changes are saved automatically when you end the session. If autosave is set to PROMPT, you are prompted to save or cancel your changes. If autosave is set to NOPROMPT, all changes since the last SAVE command are discarded. Use the AUTOSAVE or the ENDSAVE command to set autosave mode. If autosave is on and the save fails, the message "Error during save" appears. Use CREATE or REPLACE to attempt to save the file under another name. Use CANCEL to end the edit session without saving the file. The ALL operand allows you to exit the current session without processing each file in the ring separately.

ENDSAVE

The ENDSAVE command sets the autosave mode to control saving of the file when you end the edit session. It is identical to the AUTOSAVE primary command.

ENDSAVE [ON]
[OFF [PROMPT]]
[NOPROMPT]

Autosave mode has an effect only when data in the file has been changed. Data is considered to be changed when you actually modify the data, even if you later change it back to its original state. Data is also considered to be changed when you execute a command that has the potential for change, even if no change results (such as shifting a blank line or changing a string to itself). You may type command operands in any order.

ENDSAVE with no operands is identical to ENDSAVE ON. When autosave mode is on, the file is automatically saved when you end the edit session using the END primary command (normally PF3).

ENDSAVE OFF is identical to ENDSAVE OFF PROMPT. In autosave prompt mode, a prompt appears when you end the session. You may then enter SAVE to save your changes or CANCEL to terminate the session without saving.

ENDSAVE OFF NOPROMPT turns autosave mode off and suppresses prompting. In this case, END behaves exactly like CANCEL. To save changes, you **must** enter a SAVE command before pressing PF3.

EXCLUDE

The EXCLUDE command excludes from the display one or more lines that contain a specific string.

```
EXCLUDE   s1   [r] [NEXT]    [CHARS]   [c1 [c2]]
EX                [ALL]     [PREFIX]
X                 [FIRST]   [SUFFIX]
                  [LAST]    [WORD]
                  [PREV]
```

You may specify the command operands in any order. When you specify only the required string operand, EXCLUDE searches for ***s1*** within the column boundaries specified by the current BOUNDS setting. It searches for the next occurrence of ***s1***, beginning at the current cursor position. It treats ***s1*** as a character string (CHARS), meaning that it will find the string "CAN" as a single word or as a substring of other words or data as in "can", "candy", "pelican", or "decanter". It searches the entire file to locate ***s1***. It excludes the line containing ***s1*** from the current display. Excluded lines are identified by "shadow" lines that look like

```
——- n LINE(S) NOT DISPLAYED ——-
```

where ***n*** is the number of lines excluded from the display.

If you use the ALL operand to exclude all lines that contain ***s1***, the resulting display may contain several of these "shadow" lines.

The sections that follow describe how to specify strings and how to override the default actions of the EXCLUDE command.

String Operands

s1 is the search string. This operand is required.

The string operands may be any of the following types of strings:

- simple strings
- delimited strings
- character strings
- hexadecimal strings
- picture strings
- asterisk strings

Simple strings are any combination of alphanumeric characters that is not enclosed in quotes. They may not contain blanks, commas, or asterisks. A simple string is not case sensitive. An example of an EXCLUDE command using simple strings is

```
EXCLUDE ABC
```

This command finds the next occurrence of "abc", regardless of the case in which "abc" appears in the data, and excludes that line from the display.

Delimited strings are similar to simple strings but may contain blanks, commas, quotes, or asterisks. To insure that the string is processed properly, it must be enclosed in single or double quotes. If the string contains one form of quotes, it must be delimited by the other form. Use delimited strings to reference a null string. Examples of EXCLUDE commands that use delimited strings are

```
EX 'Once upon a time'
EX "It's grand"
```

Character strings provide case sensitivity. A character string is enclosed in quotes and immediately preceded or followed by the letter "C". Examples of EXCLUDE commands with character strings are

```
X C'Have a nice day'
X 'So long'C
```

Hexadecimal strings allow you to manipulate data which may not be represented by normal printing char-

acters. A hexadecimal string is enclosed in quotes and immediately preceded or followed by the letter “X”. Because each byte of data is represented by two hexadecimal digits, hexadecimal values must be entered in pairs. The examples that follow exclude lines containing a tab character and a line feed character, respectively.

```
X '09'X
EX X'0A'
```

Picture strings allow you to specify a template for the search string which may be matched by many different strings in the data. A picture string is enclosed in quotes and immediately preceded or followed by the letter “P”. Picture strings may contain a combination of normal text characters and symbolic characters. The symbolic characters used to define the template are

=	any character
^	any non-blank character
.	any non-ASCII character (except the tab character, hexadecimal 09, which is treated as a displayable character)
#	any numeric character
-	any non-numeric character
@@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character (not numeric or alphabetic)

Examples of picture strings are

`P'N=='`	the letter N followed by any two characters; matches now, new, net, n43, n&y

`P'N@@'`	the letter N followed by any two alphabetic characters; matches now, new, and net but not n43 or n&y
`P'@##^=<'`	an alphabetic character followed by two numbers, any non-blank character, any character, and any lowercase character; matches “a23aAb”, “a23A b”, and “Z99xxx” but not “a23 Ab” or “Z99xxX”.

Asterisk strings allow you to reference strings previously used in a FIND, CHANGE, or EXCLUDE command. If you specify ***s1*** as a single asterisk, that operand takes its value from the corresponding operand in the most recent FIND, CHANGE, or EXCLUDE command issued.

Range Operand

Use the range operand, ***r***, to limit the portion of the file to be searched for ***s1***. ***r*** must be specified as labels in the form “*.label1 .label2*”. If you specify user-defined labels, the labels must have been defined prior to issuing the EXCLUDE command. You may also specify a system-defined label, such as .zfirst or .zlast.

Directional Operands

Directional operands control the direction of the search for ***s1***. If no directional operand is specified, the search begins at the current cursor position and moves forward. This is identical to specifying the NEXT operand. As alternatives, you may specify

PREV	the first occurrence of ***s1*** prior to the current cursor position
FIRST	the first occurrence of ***s1***, starting at the top of the file and searching forward

LAST the first occurrence of ***s1***, starting at the bottom of the file and searching backward

ALL all occurrences of ***s1***

String Location Operands

These operands allow you to define how ***s1*** is to be interpreted. The default is CHARS, where the string may be a single word or a substring of a word or other data. Alternatives use the position of the string within the text to determine if the search is successful.

The position within the text is defined by the presence of a special character or a blank at the beginning or end of the string. For this purpose, a special character is any non-alphanumeric character except #, $, ‘, and @.

The alternative string location operands are

PREFIX The string must be preceded by a blank or special character. It may not be followed by a blank or special character.

SUFFIX The string must be followed by a blank or special character. It may not be preceded by a blank or special character.

WORD The string must be preceded and followed by a blank or special character.

The following sample line from a file is used to illustrate the string location operands. The string found is shown in bold.

PREFIX CAN ***CAN***DY PELICAN DECANTER 'CAN'
*****CAN***NOT (AFRICAN)

SUFFIX CAN CANDY PELI***CAN*** DECANTER 'CAN'
*CANNOT (AFRI***CAN**)

WORD ***CAN*** CANDY PELICAN DECANTER '***CAN***'
*CANNOT (AFRICAN)

Column Operands

These operands allow you to restrict the search for ***s1*** to specific columns of data. If only ***c1*** is specified, the search is restricted to columns between ***c1*** and the right boundary column from the current BOUNDS setting.

EXPAND

The EXPAND command converts tab characters into strings of blanks.

EXPAND

EXPAND removes the tab characters and replaces them with an equivalent number of blank spaces. It uses the current hardware tab settings to determine the number of blanks required.

Use the COMPRESS command to convert strings of blanks to tab characters.

Refer to the VIEW primary command for information on controlling the display of data containing tab characters.

FIND

The FIND command searches for a string.

```
FIND  s1  [r]  [NEXT]   [CHARS] [X]   [c1 [c2]]
F             [ALL]          [PREFIX] [NX]
              [FIRST]  [SUFFIX]
              [LAST]   [WORD]
              [PREV]
```

You may specify the command operands in any order. When you specify only the required string operand, FIND searches for ***s1*** within the column boundaries specified by the current BOUNDS setting. It searches for the next occurrence of ***s1***, beginning at the current cursor position. It treats ***s1*** as a character string (CHARS), meaning that it will find the string "CAN" as a single word or as a substring of other words or data as in "can", "candy", "pelican", or "decanter". It searches the entire file to locate ***s1***. The sections that follow describe how to specify strings and how to override the default actions of the FIND command.

If you use the ALL operand to find multiple occurrences of a string, the cursor is positioned on the first occurrence of the string. A message tells you how many occurrences of the string were found. If ***s1*** is found in an excluded line, that line is automatically re-displayed.

String Operands

s1 is the search string. This operand is required.

The string operands may be any of the following types of strings:

- simple strings
- delimited strings
- character strings
- hexadecimal strings
- picture strings
- asterisk strings

Simple strings are any combination of alphanumeric characters that is not enclosed in quotes. They may not contain blanks, commas, or asterisks. A simple string is not case sensitive. An example of a FIND command using simple strings is

```
FIND ABC
```

This command finds the next occurrence of “abc”, regardless of the case in which “abc” appears in the data.

Delimited strings are similar to simple strings but may contain blanks, commas, quotes, or asterisks. To insure that the string is processed properly, it must be enclosed in single or double quotes. If the string contains one form of quotes, it must be delimited by the other form. Examples of FIND commands that use delimited strings are

```
FIND ‘Once upon a time’
FIND “It’s grand”
```

Character strings provide case sensitivity. A character string is enclosed in quotes and immediately preceded or followed by the letter “C”. Examples of FIND commands with character strings are

```
F C’Have a nice day’
F ‘So long’C
```

Hexadecimal strings allow you to manipulate data which may not be represented by normal printing characters. A hexadecimal string is enclosed in quotes and immediately preceded or followed by the letter “X”. Because each byte of data is represented by two hexadecimal digits, hexadecimal values must be entered in pairs. The examples that follow find a tab character and a line feed character, respectively.

```
F ‘09’X
F X’0A’
```

Picture strings allow you to specify a template for the search string which may be matched by many different strings in the data. A picture string is enclosed in quotes and immediately preceded or followed by the letter "P". Picture strings may contain a combination of normal text characters and symbolic characters. The symbolic characters used to define the template are

=	any character
^	any non-blank character
.	any non-ASCII character (except the tab character, hexadecimal 09, which is treated as a displayable character)
#	any numeric character
-	any non-numeric character
@@	any alphabetic character
<	any lowercase alphabetic character
>	any uppercase alphabetic character
$	any special character (not numeric or alphabetic)

Examples of picture strings are

`P'N=='`	the letter N followed by any two characters; matches now, new, net, n43, n&y
`P'N@@'`	the letter N followed by any two alphabetic characters; matches now, new, and net but not n43 or n&y
`P'@##^=<'`	an alphabetic character followed by two numbers, any non-blank character, any character, and any lowercase character; matches "a23aAb", "a23A b", and "Z99xxx" but not "a23 Ab" or "Z99xxX".

Asterisk strings allow you to reference strings previously used in a FIND, CHANGE, or EXCLUDE command. If you specify ***s1*** as a single asterisk, that operand takes its value from the corresponding operand in the most recent FIND, CHANGE, or EXCLUDE command issued.

Range Operand

Use the range operand, ***r***, to limit the portion of the file to be searched for ***s1***. ***r*** must be specified as labels in the form "*.label1 .label2*". If you specify user-defined labels, the labels must have been defined prior to issuing the FIND command. You may also specify system-defined labels, such as .zfirst or .zlast.

Directional Operands

Directional operands control the direction of the search for ***s1***. If no directional operand is specified, the search begins at the current cursor position and moves forward. This is identical to specifying the NEXT operand. As alternatives, you may specify

PREV	the first occurrence of ***s1*** prior to the current cursor position
FIRST	the first occurrence of ***s1***, starting at the top of the file and searching forward
LAST	the first occurrence of ***s1***, starting at the bottom of the file and searching backward
ALL	all occurrences of ***s1***

String Location Operands

These operands allow you to define how ***s1*** is to be interpreted. The default is CHARS, where the string may be a single word or a substring of a word or other data. Alternatives use the position of the string within the text to determine if the search is successful. The position within the text is defined by the presence of a special character or a blank at the beginning or end of the string. For this purpose, a special character is any non-alphanumeric character except #, $, ‘, and @.

The alternative string location operands are

PREFIX The string must be preceded by a blank or special character. It may not be followed by a blank or special character.

SUFFIX The string must be followed by a blank or special character. It may not be preceded by a blank or special character.

WORD The string must be preceded and followed by a blank or special character.

The following sample line from a file is used to illustrate the string location operands. The string found is shown in bold.

PREFIX CAN ***CAN***DY PELICAN DECANTER ‘CAN’
*****CAN***NOT (AFRICAN)

SUFFIX CAN CANDY PELI***CAN*** DECANTER ‘CAN’
*CANNOT (AFR***I*****CAN***)

WORD

CAN CANDY PELICAN DECANTER '***CAN***'
*CANNOT (AFRICAN)

Scope Operands

These operands allow you to restrict the search for ***s1*** to excluded (X) or non-excluded (NX) lines.

Column Operands

These operands allow you to restrict the search for ***s1*** to specific columns of data. If only ***c1*** is specified, the search is restricted to columns between ***c1*** and the right boundary column from the current BOUNDS setting.

HEX

The HEX command is used to enable or disable hexa-decimal display of data in the file being edited.

HEX **[ON]**
OFF

HEX with no operands is identical to HEX ON. Only the vertical display mode is supported at this time. In this mode, three lines appear on the screen for each data line. The first is the normal character representation of the data. The next two lines are a vertical representation of the hexadecimal value corresponding to each character.

When hexadecimal mode is on, you may modify data by typing directly over either the character or the hexadecimal display. This allows you to change data which cannot be displayed in character format.

HEX OFF disables hexadecimal mode, returning the display to character representation only.

When you set HEX ON in conjunction with VIEW EXPAND, uni-SPF automatically changes the display to VIEW COMPRESS. This is necessary to ensure that the character and hexadecimal values align correctly in the display. If you subsequently set HEX OFF, the display returns to VIEW EXPAND.

The hexadecimal mode setting is saved with the file.

IMACRO

The IMACRO command sets the name of an initial macro to be run before the file is displayed in an edit session. The setting of IMACRO is saved in the edit profile.

IMACRO [*macro-name*]
[NONE]

macro-name is the name of the initial macro to be associated with this profile. When IMACRO is set to NONE, no initial macro is executed; this is the default.

If an initial macro is specified on the Edit-Entry Panel, it takes precedence over any IMACRO setting in the edit profile.

LEFT

The LEFT command moves the display toward the left-most column of data. It is normally associated with the PF10 key.

LEFT **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, LEFT moves left by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the LEFT command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with LEFT (normally, PF10).

During an edit session, the settings of boundary columns affect the behavior of the LEFT command. Refer to the BOUNDS command for a complete discussion of setting column boundaries and of the effects of bounds on scrolling.

Examples:

`LEFT`	scrolls left in the file by the amount indicated in the SCROLL field
`LEFT 12`	scrolls left 12 columns
`LEFT M`	repositions the display so that the leftmost column appears in column 1

LOCATE

The LOCATE command performs a directed scroll of the data display.

LOCATE *line-number*
LOC *label*
L

You must specify either a ***line-number*** or a ***label***. LOCATE repositions the display so that the ***line-number*** or ***label*** specified is the top line of data.

If you specify a user-defined label, the label must have been defined prior to issuing the LOCATE command. You may also specify a system-defined label, such as .zfirst or .zlast.

If you specify a line-number, it is relative to the beginning of the file.

MOVE

The MOVE command moves data from a disk file into the file that is currently being edited. It removes the disk file after the data is copied.

MOVE [*filename*] [BEFORE *label*]
[AFTER *label*]

You may specify the command operands in any order.

filename is the name of the file from which data is to be retrieved. If you specify only a ***filename***, uni-SPF assumes that the file is located in the current working directory. You may specify a full path name to identify the file to be retrieved. Omit this operand to display the Edit-Move panel.

If the file you are editing is not empty, you must specify a destination for the data that is retrieved. The simplest way to do this is to use the A (after) or B (before) line command. Alternatively, use the `BEFORE ***label***' or `AFTER ***label***' operand of the MOVE command. If *label* is a user-defined label, it must have been defined prior to the move. ***label*** may also be a system-defined label, such as .zfirst or .zlast. Note that the MOVE command does not recognize the optional numeric specification of the A or B line command.

If you do not specify a destination, the message "MOVE pending" appears, and your MOVE command remains on the command line. You may provide a destination using a line command, modify your command to add the appropriate operand, or erase the command to cancel its execution.

If you do not specify a ***filename***, the Edit-Move panel appears. On this panel you may provide the directory path and file name to be retrieved. The entire file is copied into the file currently being edited at the destination specified. Unlike the COPY command, MOVE does not allow you to limit the range of lines to be re-

trieved. This is because the entire file is deleted after the data is moved.

Examples:

`MOVE FILE1`

moves all data from the file named FILE1 from the current working directory into the file being edited. If the file being edited already contains data, you must use the A (after) or B (before) line command to indicate the destination for the move. FILE1 is deleted after the data is moved.

`MOVE BEFORE .ZLAST`

establishes the destination for the data to be retrieved. The Edit-Move panel appears for you to identify the file that is the source for the data. The file you specify is removed after the data is retrieved into the file being edited.

`MOVE FILE2 AFTER .NSERT`

moves data from the file named FILE2 in the current working directory into the file being edited. The contents of FILE2 are inserted after the line labeled .NSERT, which has previously been defined. FILE2 is removed.

`MOVE BEFORE .ZLAST /home/owner/subdir/myfile`

moves data from /home/owner/subdir/myfile and places it immediately preceding the last data line in the file currently being edited. /home/owner/subdir/myfile is then removed.

OVERWRITE

The OVERWRITE command controls the action taken when a file is saved in an edit session. The save may occur as a result of an explicit SAVE command or by default if ending an edit session with AUTOSAVE ON.

OVERWRITE ON
OFF

ON specifies that a save operation writes directly to the original file on disk, bypassing the temporary file created to guard against loss of data in file-system-full conditions.

OFF specifies that a save operation creates a temporary copy of the data before saving the permanent file. This is the default.

The setting of OVERWRITE is saved in the edit profile.

For additional information about safeguards against data loss when saving files, see the earlier section entitled “Protection Against Loss of Data”.

PASTE The PASTE command retrieves data from the cut buffer.

PASTE

You must indicate the destination for the data with the A or B line command. Because the data remains in the cut buffer until replaced with another CUT command, you may

- paste data in several different locations in the current file
- paste data in one or more files in the ring
- paste data cut from one edit session into another session created by SPLIT
- paste data into one or more files in a subsequent edit session. This assumes, of course, that you have not exited uni-SPF before returning to the editor.

The cut buffer is stored in each user's HOME directory in a file named ".spf.clipboard". You must have read permission for this file.

PROFILE

The PROFILE command allows you to

- display the current edit profile
- control the appearance of the display
- switch to a different profile
- define a new profile
- lock or unlock the current profile

There are two forms of the PROFILE command.

PROFILE [*name*] [*number*]
PROF
PRO
PR

In this form, both operands are optional and may be entered in any order. With no operands, PROFILE displays a minimum of four lines of the current profile at the top of the data currently being edited. If the BOUNDS or TABS are not set to the system defaults, PROFILE without operands displays the BOUNDS and/or TABS line and the COLS line.

The ***name*** operand allows you to switch to a different profile or to create a new profile. If you specify a name that does not already exist, a new profile is created. The initial settings in the new profile are taken from the profile in effect at the time you enter the PROFILE command.

The ***number*** operand controls the number of lines of the profile that appear in the display. The first four lines show the current mode settings. Lines five, six, and seven show the current TABS and BOUNDS settings and the COLS line, respectively. The MASK line is not currently supported.

PROFILE 0 resets the display to eliminate the profile information. The RESET command also performs this function.

PROFILE **LOCK**
PROF **UNLOCK**
PRO
PR

The second form of the PROFILE command allows you to lock or unlock a profile. When a profile is locked, any changes you make to a mode or option affect only the current edit session. The profile is not permanently altered. When a profile is unlocked, any changes made during the current session are saved automatically.

The default setting for PROFILE is UNLOCK. Use PROFILE LOCK to save and lock a profile.

Examples:

`PROF`	displays the first four lines of the profile at the top of data
`PROF DOC 7`	switches from the default profile to the DOC profile and displays all seven lines of the new profile at the top of data
`PROFILE 0`	removes the profile display from the screen

RCHANGE

The RCHANGE command repeats the last CHANGE command. It is normally used from a PF key and is normally assigned to PF6.

RCHANGE

The RCHANGE command uses the operands specified on the most recently issued CHANGE command.

REPLACE

The REPLACE command creates a new file from data in the file currently being edited. Either all or a portion of the current file may be written to the new file. REPLACE is identical to CREATE except that it overwrites the output file if it already exists.

REPLACE [*filename*] [*range*]
REPL
REP

The operands of the REPLACE command must be specified in the order shown.

filename is the name of the file to be created. If you specify only a filename, uni-SPF assumes that the file is to be placed in the current working directory. You may specify a full path name to identify the file to be created. Omit this operand to display the Edit-Replace panel.

You must specify a range to be written to the new file, even if you want to use all the data in the file currently being edited. This is because REPLACE differentiates between copying data to the new file and moving data from the current file to the new file. The easiest way to designate the range to be written is to use the appropriate C (copy) or M (move) line command. C, C***n***, ***n***C, or CC designate lines to be copied to the new file. M, M***n***, ***n***M, or MM designate lines to be moved to the new file and deleted from the current file.

Alternatively, you may specify the range as an operand of the REPLACE command. ***range*** must be specified using labels in the form "*.label1 .label2*". If you specify user-defined labels, the labels must have been defined prior to issuing the REPLACE command. You may also specify a system-defined label, such as .zfirst or .zlast. Using the ***range*** operand copies data to the new file.

If you do not specify a ***filename***, the Edit-Replace panel appears. On this panel you may provide the directory path and file name to be copied.

Examples:

`REPLACE VERSION2`

creates a file named VERSION2 in the current working directory. If VERSION2 already exists, its contents are entirely replaced. Copies lines marked in the current file with the C line command or moves lines marked in the current file with the M line command.

`REPLACE .ZFIRST .ZLAST`

creates or replaces a file in the current working directory. All lines of the current file are copied. The Edit-Replace panel appears for you to identify the file to be created or replaced.

`REPLACE /home/subdir/version3 .START .STOP`

creates or replaces a file named version3 in the directory /home/owner/subdir. Lines are copied from the current file beginning at the label .start and ending with the label .stop.

RESET

The RESET command resets special lines, flagged lines, and pending line commands by clearing the display. Either all or a portion of the current file may be reset.

RESet [LABel] [COMmand] [ERRor] [CHanGe] [SPEcial] [eXcluded]

You may specify multiple operands for the command; and you may enter the operands in any order.

Without operands, the command resets all options except LABel. Use appropriate combinations of operands to limit reset as shown below:

LABel	**clear user-defined labels**
COMmand	**clear pending line commands**
ERRor	**clear error flags (==ERR>)**
CHanGe	**clear change flags (==CHG>)**
SPEcial	**clear special lines (=PROF>, =TABS>, =BNDS>, =COLS>, ==MSG>)**
eXcluded	**re-display excluded lines**

Since RESET scans the entire file for conditions to be reset, it may be faster to use the D (delete) line command when you need only to remove a few special lines.

Examples:

`RESET`	clears all flagged conditions in the entire file except user-defined labels
`RES COM`	clears all pending line commands
`RES X SPE`	removes special lines from the display and re-displays excluded lines

RFIND

The RFIND command repeats the last FIND command. It is normally used from a PF key and is normally assigned to PF5.

RFIND

The RFIND command uses the operands specified on the most recently issued FIND command.

RIGHT

The RIGHT command moves the display toward the rightmost column of data. It is normally associated with the PF11 key.

RIGHT **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, RIGHT moves right by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the RIGHT command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with RIGHT (normally, PF11).

During an edit session, the settings of boundary columns affect the behavior of the RIGHT command. Refer to the BOUNDS command for a complete discussion of setting column boundaries and of the effects of bounds on scrolling.

Examples:

`RIGHT`	scrolls right in the file by the amount indicated in the SCROLL field
`RIGHT 12`	scrolls right 12 columns
`RIGHT P`	scrolls right by the number of columns that normally appear in the current edit window

SAVE

The SAVE command saves the current file without ending the edit session.

SAVE
SAV
S

SAVE replaces the existing disk file with the version currently being edited. If a SAVE cannot be completed because of an I/O error or insufficient disk space, a message indicates the nature of the error and whether or not a temporary copy of the file has been successfully saved. Use CREATE or REPLACE to save the file under a different name. Use CANCEL to end the edit session without saving the file.

SHELL

The SHELL command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The SHELL command is synonymous with the ! command.

SHELL [*string*]

With no operands, SHELL takes you to the Unix shell. The specific shell used is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press Enter when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. If the Unix command displays information on the screen, the message

```
Press ENTER to continue uni-SPF
```

appears after the output of the Unix command.

SORT

The SORT command reorders data in the file you are editing. SORT always reorders full records.

SORT [*range*] [*sort-field-1* [. . . *sort-field-n*]]

You may specify the command operands in any order.

With no operands, SORT reorders the entire file in ascending order.

The ***range*** operand specifies the beginning and ending lines for the sort. To distinguish the ***range*** from the sort fields, you must use labels to specify the range. If you use user-defined labels, the labels must have been defined prior to issuing the SORT command. You may also use system-defined labels such as .zfirst or .zlast.

The ***sort-field*** operands define the columns to be compared for reordering data. You may specify any number of sort fields. If no sort fields are specified, the default is the current BOUNDS setting. Each sort field has the form

[*sort-order*] [*start-column* [*end-column*]]

The ***sort-order*** operand is either A or D and indicates whether the field associated with it should be sorted in ascending or descending order. If this operand is omitted, the default of ascending order is assumed. You may use this operand with SORT or SORT ***range*** to control the sort order when no sort fields are specified.

start-column is the number of the column in which this sort field begins. ***end-column*** is the number of the column in which this sort field ends. If ***end-column*** is omitted, the default is the current right BOUNDS. ***start-column*** must precede ***end-column***; and neither value may be outside the current BOUNDS. Sort fields may not overlap.

Each sort field may have a ***sort-order*** associated with it. If no ***sort-order*** is specified, ascending is assumed.

The ***sort-order*** operand may precede or follow the sort field specification. The following sort commands produce identical results:

```
SORT A 1 10 30 35
SORT 1 10 A 30 35
SORT A 1 10 A 30 35
SORT 1 10 A A 30 35
SORT 1 10 A 30 35 A
SORT A 1 10 30 35 A
SORT 1 10 30 35
```

In the current release, the ***sort-order*** must be the same for all sort fields. If it is not, an error message appears.

Sorting is performed according to the hexadecimal representation of the data. Use the HEX primary command to display the hexadecimal representation of data in the file you are editing.

Examples:

`SORT` sorts the entire file in ascending order

`SORT D` sorts the entire file in descending order

`SORT .ZFIRST .TEN D`
reorders lines from the beginning of the file to the line labelled .TEN in descending order

`SORT 25 30`
sorts the entire file in ascending order based on the contents of data in columns 25-30

`SORT D 55`
sorts the entire file in descending order based on the contents of data between column 55 and the current right BOUNDS

```
SORT 55 A 1 10
```

sorts the entire file in ascending order using data between column 55 and the current right BOUNDS as the primary sort field and data in columns 1 through 10 as the secondary sort field

```
SORT .MID .ZLAST 14 20 A 60
```

sorts lines between the line labelled .MID and the end of the file using data in columns 14 through 20 as the primary sort field and data between column 60 and the current right BOUNDS as the secondary sort field

SUBMIT

The SUBMIT command is used to submit for execution the contents of the file currently being edited. It is valid only from the editor command line.

SUBmit [BATCH]
[BG [*filename*]]
[FG [FILE] [*filename*]]

If no keywords are specified, the default is BATCH.

If BATCH is specified, the job is submitted to the Unix batch queue using the "batch" command. It is run when system load permits and the system sends email when the job is complete. All output from the job should be directed, within the job, to a file. Batch jobs are logged and tracked in the Jobs Status table, accessible from Option 3.J.

If BG is specified, the job is submitted in the background using the Unix syntax ***command &*** .

If ***filename*** is omitted, SUBMIT captures stdout and stderr from the job in a file in the directory $HOME/.ISPsubmit using the filename ***jobname***.out, where ***jobname*** is the system- or user-assigned job name for this submission.

If ***filename*** is specified, SUBMIT captures stdout and stderr from the job in the file designated.

Background jobs are logged and tracked in the Jobs Status table, accessible from Option 3.J.

If FG is specified, the job is submitted in the foreground using the Unix syntax ***command***.

If the FILE keyword is omitted, output is directed to the screen; work in uni-SPF is suspended until the executing job is complete.

If FILE is supplied without the optional ***filename***, SUBMIT captures stdout and stderr from the job in the same way as for a background submission in a file in the directory $HOME/.ISPsubmit. If the token following "FG" is not "FILE" (any case), then it is assumed to be the name of a file in which stdout and stderr from the job are captured. In the case of "FILE" or a ***filename***, the output file is automatically added to the Edit ring upon completion of the job and displayed for the user. Because foreground jobs run immediately and within the context of the current uni-SPF session, they are not logged or tracked through the Jobs Status table.

In the absence of a JOB card that includes a "TO=" specification, SUBMIT determines what processor to send the job to according to the following priorities:

(1) If the file contains an implicit execution string (such as #!/bin/sh or #!/usr/local/rexx/rxx), then the job is submitted there; if more than one such string exists in the file, the first encountered is the one used.

(2) If no implicit execution string is found, the file is searched for a REXX-style comment that includes the characters `rexx' in any case; such a comment must be fully contained on a single line as in

```
/* run this using uni-REXX */
```

or

```
/* REXX */
```

This is to support REXX programs that may have been ported from the mainframe. If such a comment is found, the job is sent to uni-REXX.

(3) If neither (1) nor (2) is found, the job is submitted to the processor

identified by the user's SHELL environment variable.

The file being edited – i.e., the job to be submitted – may have a JOB card that provides certain information about the job to be run. The format of the JOB card is

#JOB [TYPE=BATCH|BG|FG] [TO=*command-processor*]
[NAME=*jobname*] [FILE=*filename*]

The job card parameters must appear on a single line in the file. They may appear in any order and any of them may be omitted. The job card is discarded before the job is actually run. Any operand specified on the command line overrides its counterpart on a JOB card.

TYPE=
specifies whether this should be a batch or background submission, as described above for the command line operand; if BATCH, BG, or FG is specified as an operand of the SUBMIT command, it and any additional operands supplied override any specification for TYPE or FILE that may appear on the job card

TO=
specifies the command processor to which this job is to be submitted; typically, this will be something like "csh", "sh", "ksh", "rxx", or "make"; if this parameter is omitted, SUBMIT determines what processor to send the job to according to the priorities described above for command-line execution of SUBMIT

NAME=
specifies a unique job name for this submission; the ***jobname*** is used for the temporary files created in $HOME/.ISPsubmit and for the output file captured for background jobs; if NAME= is omitted a name is supplied by SUBMIT of the form

JOB*nnnn* where ***nnnn*** is a unique number determined by adding 1 to the highest number present in the Job Status table

FILE=
specifies the name of the file in which output from a foreground or background job is captured; the name specified may be a filename or a full pathname to the file; if it is a filename, it is written to the current working directory

A job card may be commented out. This is done in the comment style for the language of the job. If the job is a shell script or a make file, the job card is commented out by preceding it with an additional # sign (or, alternatively, overtyping the "J" with a "#", in which case the #JOB syntax would not be recognized by the SUBMIT command). If the job is a uni-REXX program, the job card is commented out by enclosing it in REXX-style comment symbols (/* .... */).

When a job is submitted, its submission parameters from the JOB card or from system-determined defaults are saved in a uni-SPF table named ISPJOBS.tbl in the user's HOME directory. The Job Status Utility (Option 3.J) displays the contents of this table.

Foreground jobs are not tracked through the Jobs Status table. Foreground jobs are run immediately. If output is to the screen, the uni-SPF display is replaced by the job output until execution of the job is complete. If output is to a file, uni-SPF activity is "locked" until completion of the job, at which time the output file is automatically added to the Edit ring and displayed for the user. The temporary files that must be created to run each job are placed in a directory named .ISPsubmit in the user's HOME directory. This directory is created for you the first time you execute a SUBMIT command.

TABDEF

Use the TABDEF command to restore tab stops to the defaults for the current file.

TABDEF

If the filename extension of the current file matches one of the language-specific profiles that are standard in uni-SPF, TABDEF restores tab stops to the defaults for that profile. Otherwise, TABDEF sets tab stops every eight columns. This is equivalent to typing an asterisk in every eighth column on the TABS definition line.

TABS

Use the TABS command to enable or disable hardware tabs.

TABS ON
OFF

When TABS are OFF, the tab key moves the cursor to the beginning of the next input field. When TABS are ON, the tab key moves the cursor automatically to each tab stop in the data as well as to the beginning of each input field.

The current TABS setting is saved in the profile.

UP

The UP command moves the display toward the top of the file. It is normally associated with the PF7 key.

UP **[Page]**
[Half]
[Csr]
[Data]
[Max]
[*n*]

When no operand is specified, UP moves up by the amount specified in the SCROLL field in the upper right corner of the screen. ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of scrolling, including how to change the value of the SCROLL field. Use one of the operands of the UP command to override the SCROLL field value.

You may use these operands with a PF key by typing the operand on the command line and pressing the PF key associated with UP (normally, PF7).

Examples:

`UP`	scrolls toward the top of the file by the amount indicated in the SCROLL field
`UP 12`	scrolls up 12 lines
`UP M`	scrolls to the top of the file

VIEW

The VIEW command controls the appearance on the screen of data that contains tab characters.

VIEW EXPAND
COMPRESS

With VIEW EXPAND, a tab character in the data automatically shifts the data to the column immediately following the next tab stop. This is the default. The cursor cannot be placed in the intervening spaces as there is no real data there.

With VIEW COMPRESS, a tab character occupies a single character position in the data and no expansion occurs.

Tab stops are defined on the =TABS> line. Refer to the TABS line command and the TABDEF primary command for details on defining tab stops.

! (Exclamation Point)

The ! command executes a Unix shell command or takes you to the shell, where you may execute one or more shell commands. The ! command is synonymous with the SHELL command.

![*string*]

With no operands, ! takes you to the Unix shell. The specific shell used is determined by the SHELL environment variable. To return to the editor, type “exit” or press CTL-D and then press Enter when prompted.

If you specify a ***string***, it is passed to the Bourne shell for execution. If the Unix command displays information on the screen, the message

```
Press ENTER to continue uni-SPF
```

appears after the output of the Unix command.

Line Commands

Editor Line Commands operate on single lines or blocks of lines. The syntax of a line command is a very short abbreviation of the command. Some line commands may be immediately preceded or followed by a numeric operand indicating the number of lines on which the operation is to be performed. Line Commands are typed in the line command area of the line on which the action is intended to occur.

The edit Line Commands are

AFTER
BEFORE
BOUNDS
COPY
COLS
DELETE
FIRST
INSERT
LAST
LOWERCASE
MOVE
OVERLAY
REPEAT
SHOW
TABS
TEXT JOIN
TEXT SPLIT
UPPERCASE
EXCLUDE
(
)
>
<
.

The pages which follow contain a detailed description of each of the line commands.

A
[After]

Use After in conjunction with Move or Copy line commands or primary commands to specify the destination for the move or copy.

A
A*n*
***n*A**

The first form identifies the line after which moved or copied data is to be placed. The other two forms allow you to specify the number of times that the data is to be repeated at the location specified. An A command remains pending until a corresponding Copy or Move command is entered.

B
[Before]

Use Before in conjunction with Move or Copy line commands or primary commands to specify the destination for the move or copy.

B
B*n*
***n*B**

The first form identifies the line before which moved or copied data is to be placed. The other two forms allow you to specify the number of times that the data is to be repeated at the location specified. A B command remains pending until a corresponding Copy or Move command is entered.

BOUNDS

The BOUNDS line command adds a boundary-definition line to the display at the location specified.

BOUNDS
BOUND
BNDS
BND

BOUNDS inserts a boundary-definition line immediately above the line on which you type the command. The boundary-definition line is identified by "=BNDS>" in the line number area. Current boundary settings are indicated by "<" and ">", for left and right bounds, respectively. Changing the position of these boundary characters by typing directly on the boundary-definition line alters the current bounds settings.

Column boundaries limit the scope of the following editor operations:

- left-shift and right-shift line commands
- left and right scrolling
- FIND, CHANGE, and EXCLUDE primary commands when you do not specify explicit columns as part of the command
- O, OO line commands
- TS line command

The effect of bounds settings on scrolling is special. Scrolling to the left or right stops at the left or right bound, respectively. A subsequent scroll in the same direction scrolls beyond the bound, assuming that the bound is not the leftmost or rightmost column position.

The boundary-definition line is not part of the data and is not saved with the file. The boundary column settings are saved in the profile.

Use the RESET primary command or the D line command to remove the boundary-definition line from the display.

C
[Copy]

The Copy command copies one or more lines to another location in the current file. When used in conjunction with the CREATE or REPLACE primary command, it copies one or more lines to another file. When using Copy to duplicate lines in the current file, you must specify the destination for the copy with an After, Before, or Overlay line command.

C
C*n*
***n*C**
CC

The first form copies a single line to the specified location. The second and third forms copy ***n*** lines, beginning with the line on which you type the Copy command. A C, C***n***, or ***n***C command remains pending until a corresponding destination line command (A, B, or O) is entered.

The fourth form is used to copy a block of lines. You must type CC on both the first and the last lines of the block to be copied. This is similar to the C***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type CC at one end of the block and scroll forward or backward to locate the other end of the block and the destination of the copy. A single CC command remains pending until a matching CC is entered; and a pair of CC commands remain pending until a corresponding destination command is entered.

COLS

The COLS line command adds a column-position line to the display at the location specified.

COLS
COL

COLS inserts a column-position line immediately above the line on which you type the command. The column-position line is identified by "=COLS>" in the line number area. This line is used for position reference only. You may not type over its contents.

The column-position line is not part of the data and is not saved with the file.

Use the RESET primary command or the D line command to remove the column-position from the display.

D
[Delete]

The Delete command deletes one or more lines in the current file.

D
D*n*
***n*D**
DD

The first form deletes a single line. The second and third forms delete ***n*** lines, beginning with the line on which you type the Delete command.

The fourth form is used to delete a block of lines. You must type DD on both the first and the last lines of the block to be copied. This is similar to the D***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type DD at one end of the block and scroll forward or backward to locate the other end of the block. A single DD command remains pending until a matching DD is entered.

F
[First]

The First command displays one or more lines at the beginning of an excluded block. This command is only valid on the "shadow" line that indicates the number of lines in the excluded block.

F
F*n*
***n*F**

The first form re-displays the first line at the beginning of the excluded block. The other two forms display the first ***n*** lines at the beginning of the block.

I
[Insert]

The Insert command inserts one or more blank lines in the current file.

I
I*n*
***n*I**

The first form inserts a single line immediately following the line on which you typed the Insert command. The other two forms insert ***n*** lines.

Once you have used Insert to add lines to the file, you have the option of remaining in "continuous insert" mode until you have added all the necessary data. When you enter data on the last (or only) inserted line and leave the cursor in the data portion of that line, the editor automatically inserts another line when you press Enter. To terminate "continuous insert" mode, move the cursor to a different line (possibly the Command Line) before pressing Enter; or press Enter a second time to remove the newly inserted line.

Inserted lines are identified by ‘’’’’ in the line command area. These are temporary, unnumbered lines that do not become part of the file until data is entered in them.

L
[Last]

The Last command displays one or more lines at the end of an excluded block. This command is only valid on the "shadow" line that indicates the number of lines in the excluded block.

L
L*n*
***n*L**

The first form re-displays the last line of the excluded block. The other two forms display the last ***n*** lines of the block.

LC
[Lowercase]

The LC command converts all uppercase alphabetic characters on one or more lines to lowercase.

LC
LC*n*
***n*LC**
LCC
LCLC

The first form converts a single line. The second and third forms convert ***n*** lines, beginning with the line on which you type the LC command.

The last two forms are used to convert a block of lines. You must type LCC (or LCLC) on both the first and last lines of the block to be converted. This is similar to the LC***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type LCC (or LCLC) at one end of the block and scroll forward to locate the other end of the block. A single LCC (or LCLC) command remains pending until a matching LCC (or LCLC) command is entered.

The LC command has no effect on the CAPS mode setting.

M
[Move]

The Move command moves one or more lines to another location in the current file. When used in conjunction with the CREATE or REPLACE primary commands, it moves one or more lines to another file. When using Move to rearrange lines in the current file, you must specify the destination for the move with an After, Before, or Overlay line command.

M
M*n*
***n*M**
MM

The first form moves a single line to the specified location. The second and third forms move ***n*** lines, beginning with the line on which you type the Move command. An M or M***n*** command remains pending until a corresponding destination line command (A, B, or O) is entered.

The fourth form is used to move a block of lines. You must type MM on both the first and the last lines of the block to be copied. This is similar to the M***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type MM at one end of the block and scroll forward or backward to locate the other end of the block and the destination of the move. A single MM command remains pending until a matching MM is entered; and a pair of MM commands remain pending until a corresponding destination command is entered.

O
[Overlay]

The Overlay command identifies the target line or lines on which data to be copied or moved will be overlaid. It is used to merge data from the source and target lines.

O
O*n*
***n*O**
OO

The first form specifies a single line to be overlaid. The second and third forms specify that ***n*** lines are to be overlaid, beginning with the line on which O***n*** is typed. An O or O***n*** command remains pending until a corresponding Copy or Move command is entered.

The fourth form is used to identify a block of lines to be overlaid. You must type OO on both the first and the last lines of the block to be overlaid. This is similar to the O***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type OO at one end of the block and scroll forward to locate the other end of the block. A single OO command remains pending until a matching OO is entered; and a pair of OO commands remain pending until a corresponding Copy or Move command is entered.

When you use Overlay to specify the target of a copy or move command, characters from the source line(s) overlay blank characters on the target line(s). Existing, non-blank characters on the target line(s) are not overlaid.

Only characters within the current column boundaries participate in an overlay. Characters from the source line which are outside the bounds do not overlay the target line.

To avoid loss of any data, combinations of Move and Overlay are handled in a special manner. If all characters in the source line correspond to blanks in the target

line and are within the current bounds, the overlay occurs and the original source line is deleted as in a normal Move operation. If the source line contains characters where the target line also contains non-blank characters and all data on the source line is within the current bounds, the overlay occurs normally, but the source line is not deleted. If any data on the source line is outside the current bounds, data within the bounds overlays the target line normally, but the source line is not deleted.

It is not necessary for the number of source lines to be the same as the number of target lines. When there are more target lines than source lines, the source lines are repeated as necessary to overlay all target lines. When there are more source lines than target lines, extra source lines are ignored. When there are more source lines than target lines for a move operation, the number of source lines deleted matches the number of target lines.

R [Repeat]

The Repeat command replicates a line or block of lines one or more times in the current file.

R
R*n*
***n*R**
RR
RR*n*
***n*RR**

The first form replicates a single line. The duplicate is placed immediately after the line on which you typed the R command. The second and third forms replicate the current line ***n*** times, placing the duplicates immediately after the line on which you typed R***n***.

The last three forms are used to replicate a block of lines. RR repeats the block once. RR***n*** or ***n***RR repeats it ***n*** times. You must type RR (or RR***n***) on both the

first and the last lines of the block to be duplicated. You may type RR (or RR***n***) at one end of the block and scroll forward or backward to locate the other end of the block. A single RR command remains pending until a matching RR is entered. It is necessary to provide the ***n*** value on only one of the pair of RR commands.

S
[Show]

The Show command displays one or more lines from a block of excluded lines, based on the indentation level of each line. This command is valid only on the "shadow" line that indicates the number of lines in the excluded block.

S
S*n*
***n*S**

The first form displays the line with the least amount of indentation. The other two forms display the ***n*** lines with the least amount of indentation. If more than ***n*** lines meet this criteria, only the first ***n*** are displayed.

If the effect of the Show command would leave only one line excluded in the resulting display, then all lines in the excluded block are displayed. Thus, if you enter S on block with two lines excluded, both lines are re-displayed.

TABS

The TABS line command adds a tab-definition line to the display at the location specified.

TABS

TABS inserts a tab-definition line immediately above the line on which you type the command. The tab-definition line is identified by "=TABS>" in the line number area. Current hardware tab settings are indi-

cated by asterisks (*). To change the hardware tab stops, add or delete asterisks by typing directly on the tab-definition line. Refer to the TABDEF and TABS primary commands for more information on setting and using tab stops.

The tab-definition line is not part of the data and is not saved with the file.

Use the RESET primary command or the D line command to remove the tab-definition line from the display.

TJ [Text Join]

The TJ command joins text from two lines. Text from the next line in the file is appended to the end of the line on which you type the TJ command.

TJ

The TJ line command is, in some sense, the reverse of the TS (Text Split) line command. The two commands are often used in conjunction with each other to "re-flow" text when significant insertions are required.

The TJ command honors the current BOUNDS settings. If text to be joined would extend past the right bound, only the characters that fit within the BOUNDS are joined. The residual text remains on the following line and is shifted left by the number of characters actually joined to the line above.

If you enter TJ commands on consecutive lines, only the alternate TJ commands have any effect. The first TJ joins the following line to the current line, effectively discarding the second TJ command.

TS [Text Split]

The TS command splits a line of data at the cursor location and inserts one or more blank lines between the line segments.

TS
TS*n*
***n*TS**

In all forms, you type TS on the line you wish to split and then move the cursor to the split point. When you press Enter, the editor splits the data and inserts one or more blank lines. The first form of the command inserts one blank line. The other two forms allow you to specify the number of blank lines to be inserted. Because TS is dependent on cursor position within the data, you may wish to assign this command to a PF key.

The TS command honors the current BOUNDS settings. Text to the right of the cursor, up to and including the right column boundary, is positioned at the left column boundary on the line after the inserted line(s). Text to the right of the right bound is not moved.

Lines inserted by TS behave exactly as if they were inserted by the I line command. Specifically, “continuous insert” mode is available. Refer to the I (Insert) line command for a complete discussion of “continuous insert” mode.

UC [Uppercase]

The UC command converts all lowercase alphabetic characters on one or more lines to uppercase.

UC
UC*n*
***n*UC**
UCC
UCUC

The first form converts a single line. The second and third forms convert ***n*** lines, beginning with the line on which you type the UC command.

The last two forms are used to convert a block of lines. You must type UCC (or UCUC) on both the first and last lines of the block to be converted. This is similar to the UC***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type UCC (or UCUC) at one end of the block and scroll forward or backward to locate the other end of the block. A single UCC (or UCUC) command remains pending until a matching UCC (or UCUC) command is entered.

The UC command has no effect on the CAPS mode setting.

X [Exclude]

The eXclude command excludes one or more lines from the current display.

X
X*n*
***n*X**
XX

The first form excludes a single line. The second and third forms exclude ***n*** lines, beginning with the line on which you type the exclude command.

The fourth form is used to exclude a block of lines. You must type XX on both the first and the last lines of the block to be copied. This is similar to the X***n*** form of the command but may be more useful when the block is larger than the display screen. In such instances, type XX at one end of the block and scroll forward or backward to locate the other end of the block. A single XX command remains pending until a matching XX is entered.

< [Data Shift Left]

The < line command shifts data to the left without affecting the position of program labels or comments. This allows you to easily restructure indented text or program code.

<
<*n*
n<

<<
<<*n*
n<<

The first form shifts data two columns to the left on a single line. The second and third forms also operate on a single line but allow you to specify the number of columns that data is to be shifted.

The last three forms perform the equivalent operations on a block of lines. You must type a block shift command on both the first and last lines of the block to be shifted. A single << command remains pending until a matching << command is entered.

Data shifting is accomplished by making the following assumptions about the format of the data:

- labels begin in column 1
- data subject to shifting contains no more than one blank between tokens
- data enclosed in quotation marks is a single token and is subject to shifting, regardless of how many consecutive blanks may occur inside the quotation marks
- comments (or other data not subject to shifting) are separated from shiftable data by at least two consecutive blanks

Shifting is performed according to the following rules:

- Scanning starts at the left boundary column.
- No shifting occurs until after the first blank character is found.
- Shifting begins with the first non-blank charac- ter following the first blank.
- Shifting stops when two consecutive blank characters are encountered between tokens.
- Non-blank characters are **never** deleted or truncated.
- No data is shifted beyond the left boundary column. If the shift command would cause data to move beyond the left bound, the data is shifted up to the left bound and the line is marked with ==ERR>.

> [Data Shift Right]

The > line command shifts data to the right without affecting the position of program labels or comments. This allows you to easily restructure indented text or program code.

```
>
>>n
n>

>>
>>n
n>>
```

The first form shifts data two columns to the right on a single line. The second and third forms also operate on a single line but allow you to specify the number of columns that data is to be shifted.

The last three forms perform the equivalent operations on a block of lines. You must type a block shift command on both the first and last lines of the block to be

shifted. A single >> command remains pending until a matching >> command is entered.

Data shifting is accomplished by making the following assumptions about the format of the data:

- labels begin in column 1
- data subject to shifting contains no more than one blank between tokens
- data enclosed in quotation marks is a single token and is subject to shifting, regardless of how many consecutive blanks may occur inside the quotation marks
- comments (or other data not subject to shifting) are separated from shiftable data by at least two consecutive blanks

Shifting is performed according to the following rules:

- Scanning starts at the left boundary column.
- No shifting occurs until after the first blank character is found.
- Shifting begins with the first non-blank charac- ter following the first blank.
- Shifting stops when two consecutive blank characters are encountered between tokens.
- Non-blank characters are **never** deleted or truncated.
- No data is shifted beyond the right boundary column. If the shift command would cause data to move beyond the right bound, the data is shifted up to the right bound and the line is marked with ==ERR>.

([Column Shift Left]

The (line command shifts data one or more columns to the left. The default shift is two columns. Only data within the current bounds settings is shifted. **Data shifted left of the current left bounds is lost**.

(
(*n*
***n*(**

((
((*n*
***n*((**

The first form shifts data on the current line two columns to the left. The second and third forms shift data on the current line ***n*** columns to the left.

The last three forms allow you to identify a block of lines to be shifted. You must type ((on both the first and last lines of the block to be shifted. A single ((command remains pending until a matching ((command is entered.

) [Column Shift Right]

The) line command shifts data one or more columns to the right. The default shift is two columns. Only data within the current bounds settings is shifted. **Data shifted right of the current right bounds is lost**.

)
)*n*
***n*)**

))
))*n*
***n*))**

The first form shifts data on the current line two columns to the right. The second and third forms shift data

on the current line ***n*** columns to the right. The last three forms allow you to identify a block of lines to be shifted. You must type)) on both the first and last lines of the block to be shifted. A single)) command remains pending until a matching)) command is entered.

. [Label Assignment]

The . line command assigns a symbolic name, or label, to a line. Labels are often used in range specifications for editor primary commands.

.label

The ***label*** name may be one to five alphanumeric characters in length. However, user-defined labels may not begin with the letter "Z" since this letter is reserved for system-defined labels.

System-defined labels, which are available at all times, include:

.ZCSR	**Line on which the cursor is currently located**
.ZF	**First line of the file**
.ZFIRST	**First line of the file**
.ZL	**Last line of the file**
.ZLAST	**Last line of the file**

With the exception of system-defined labels, a line may not have more than one label assigned to it.

You may remove a label from a line in one of two ways:

- type the same name on a different line
- use Erase-EOF or the space bar to blank out the name

Use RESET LABel to remove all user-defined labels.

When using labels as range specifications within primary commands, always include the leading period in the label reference.

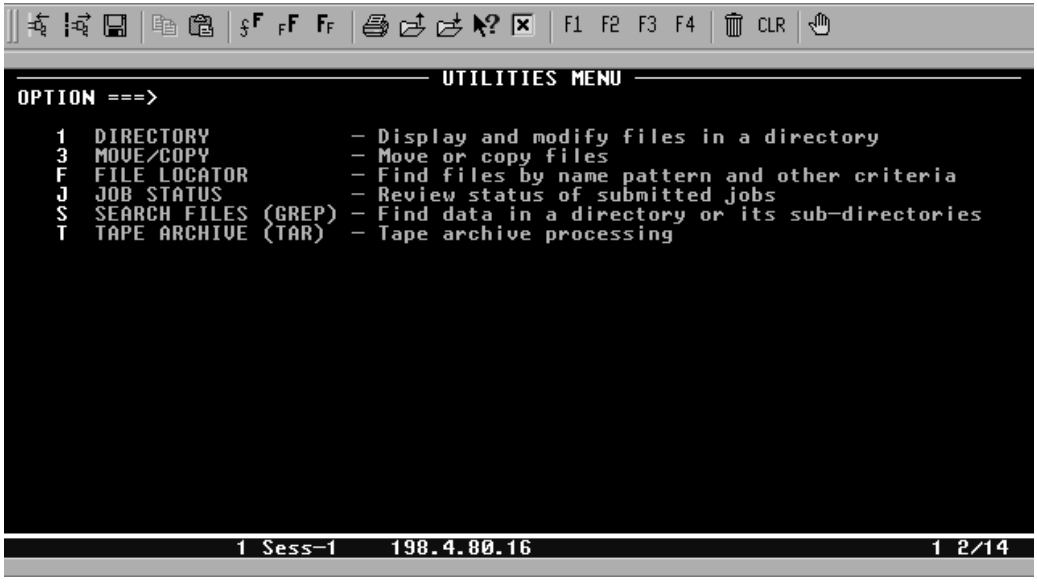
Labels are not considered part of the data and are not stored with the file.

Chapter 6: Utilities

The uni-SPF Utilities feature allows you to perform a variety of operations on directories and files. These include

- copying or moving files
- renaming files
- printing files
- deleting files
- changing file permissions
- browsing or editing files
- compressing or uncompressing files
- changing the current working directory
- creating new directories
- printing directory listings
- deleting empty directories
- locating files in the system
- searching for data within a directory tree
- monitoring the status of submitted jobs
- processing tape archives

When you select Utilities (Option 3) from the Main Menu panel, the Utilities Menu panel (shown on the following page) appears.



Select the desired utility from the menu shown.

Directory Utility (Option 3.1)

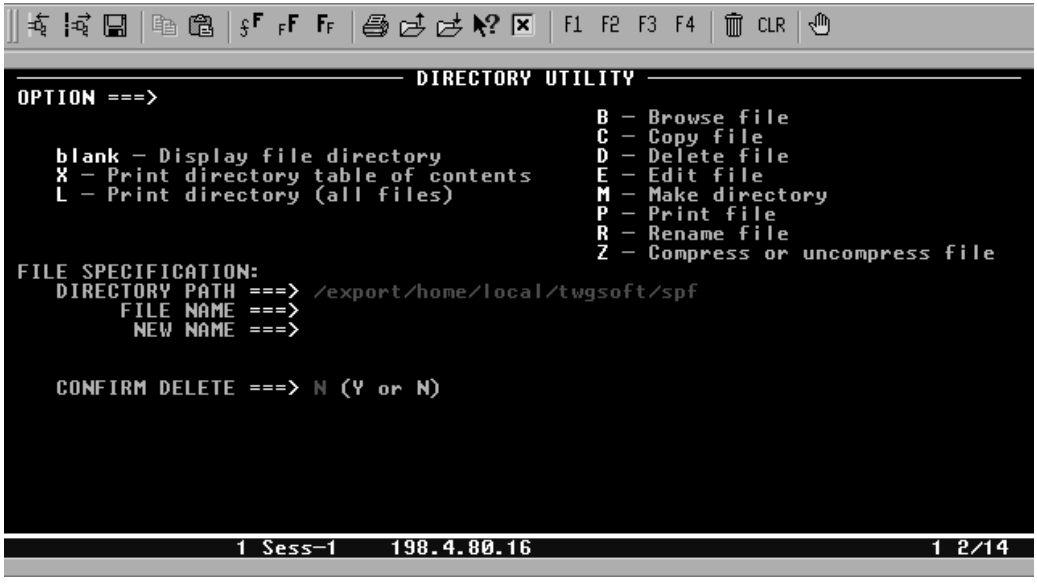
The Directory Utility allows you to perform specific operations on files or directories. For files, the operations include

- copy
- rename
- print
- delete
- edit
- browse
- change permissions
- compress or uncompress

For directories, the operations include

- create new directory
- change current directory
- print directory listing
- delete empty directory

The Directory Utility panel is shown below. You may perform the desired operation directly on this panel by completing the input fields. Available operations and the single-letter command to execute them are shown on the panel. In the File Specification fields, the Directory Path and File Name fields are automatically provided



from the most recent use of this field or from system defaults.

Type the operation you wish to perform in the primary input field. Complete or modify the information in the File Specification fields.

If the you select the Copy or Rename operation, be sure to complete the New Name field.

If you select the Make Directory operation, specify the full path of the directory to be created in the Directory Path field. If you make a new directory, it automatically becomes the current working directory.

If you select the Compress or Uncompress File option, the action taken depends on the file name. If the name of the file does **not** end in ".Z", a compress operation is performed. If the name of the file **does** end in ".Z", an uncompress operation is performed. If a file is small enough that it would not benefit from compression, this option does not compress the file but displays a message.

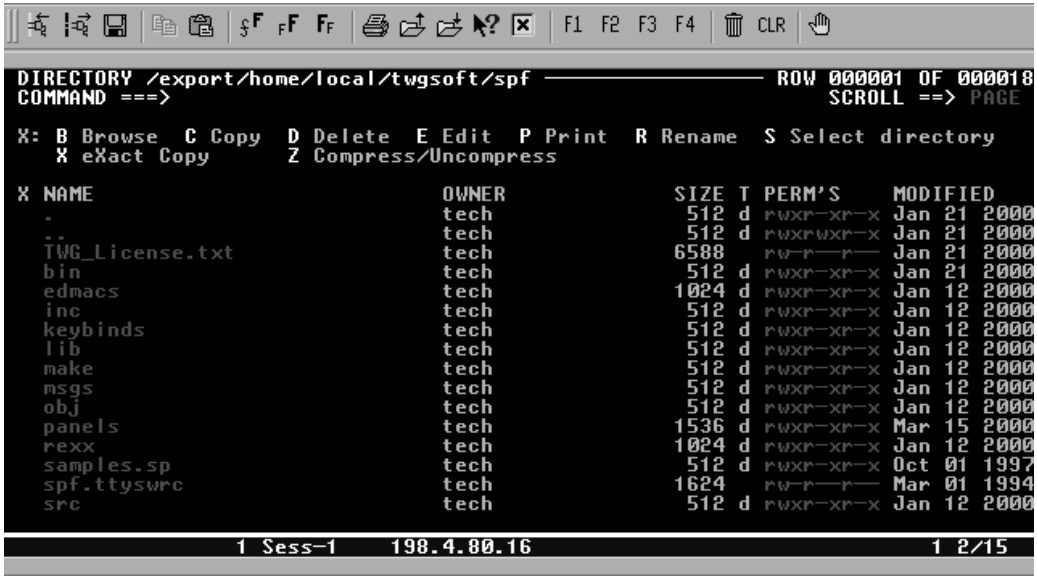
The Confirm Delete field allows you to request the display of a confirmation panel when you attempt to delete a file. The default setting of "N" suppresses the confirmation panel – delete requests are processed immediately. If you set this field to "Y", a delete command causes display of the confirmation panel. You may then press Enter to execute the delete or PF3 to abandon the operation without deleting the file.

When you press Enter, the operation is performed. uni-SPF displays a message to indicate the completion status of the operation. When both the Directory Path field and the File Name field contain information, uni-SPF assumes that you wish to perform the desired operation from the current panel. If you complete these fields without selecting an operation, the message "Enter required field" appears.

If you do not know the name of the file you wish to use or if you want to operate on multiple files, leave the primary input field blank. This results in the display of a file selection panel when you press Enter. To see all the files in the current directory, leave the File Name field blank. To restrict the listing to specific files, use Unix wild-card characters in the filename.

Chapter 2: Interaction with uni-SPF provides details on using wild-card characters.

When you press Enter, the file selection panel appears.



Normal scrolling operations are available. Automatic scrolling occurs upon successful completion of operations. The filename associated with the last successful operation automatically moves to the top of the display. Directed scrolling is also available using the LOCATE command, which has extended syntax within the Directory Utility. There are two forms of the LOCATE command.

LOCATE ***filename***
LOC
L

This form is similar to the standard syntax. If ***filename*** includes blanks, it must be enclosed in quotes.

LOCATE ***type target***
LOC
L

This form permits greater flexibility in directing the scrolling of this panel. It repositions at the top of the display the first entry that matches the location criteria specified. If ***type*** is omitted, this form of LOCATE behaves exactly like the previous form.

type may be any of the following:

FNAME **DIRECTORY** **ORIGINAL**	the complete filename
NAME **BASE**	the portion of the filename that precedes the last "." character
EXT	the portion of the filename that follows the last "." character
OWNER **OWN**	the owner of the file, as shown in the "OWNER" column of the display
SIZE **BYTES** **TRACK**	the file size, as shown in the "SIZE" column of the display
TYPE **TYP**	the type of file (directory, link, socket, etc.), as shown in the "T" column of the display
PERM'S **PERMS** **PERM**	the file permissions, as shown in the "PERM'S" column of the display
MODIFIED **DATE**	the file modification date, as shown in

the "MODIFIED" column of the display

CREATED
REFERRED

TIME the file modification time, as shown in the "MODIFIED" column of the display. This type is valid only when the third token of the MODIFIED field contains a colon (:) as in "12 Jun 14:20".

The ***target*** is a string that is appropriate for the ***type*** specified. It is permissable but not necessary to enclose ***target*** in quotes when it contains embedded blanks, as in the case of dates. With the exception of dates, the search for ***target*** is case insensitive.

Examples:

`LOCATE list`

scrolls the file selection panel so that the entry for a file whose name contains "list" as the first four characters appears at the top of the display. If no file matches this criterion, the display is positioned at the row immediately above where a filename beginning with "list" would normally appear.

`LOC EXT c`

finds the first file of the form "***name***.c" and scrolls the file selection panel so that entry appears at the top.

`L DATE Jul 4`

finds the first file whose modification date is "Jul 4" and scrolls the file selection panel so that entry appears at the top.

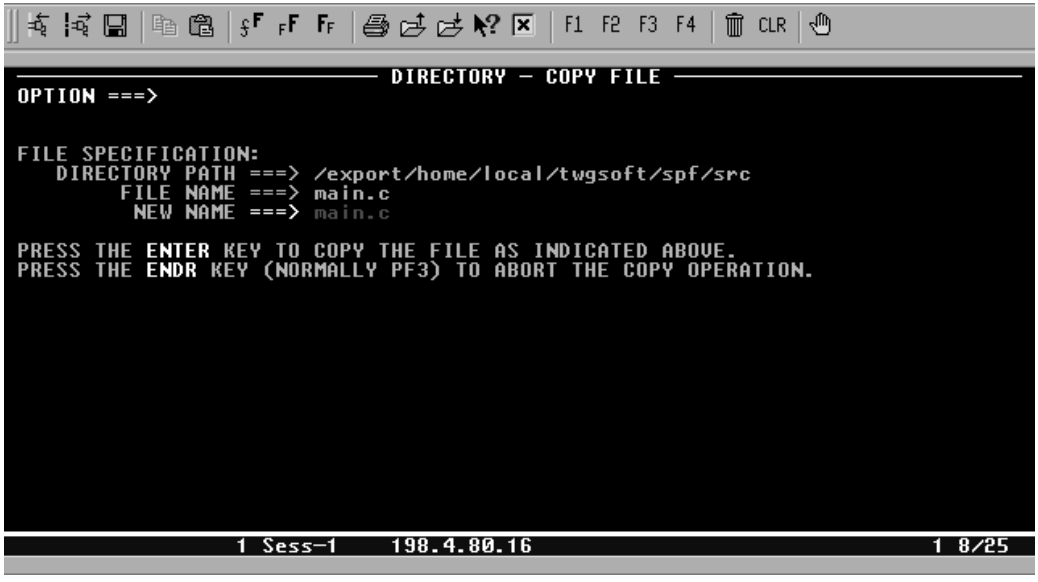
Unlike other file selection panels, there are multiple operations that you may perform and multiple input fields within the panel. The tab key moves automatically to the beginning of each input field.

The available operations are shown across the top of the panel. To perform an operation on a file, type the appropriate letter in the column labelled "X" to the left of the file name. You may type the same operation for several files or multiple operations on different files. When you press Enter, all operations are performed. An asterisk (*) in the "X" column indicates successful completion of the operation. A question mark (?) in the "X" column indicates failure of the operation.

If you type multiple operations, they are performed in the order in which they occur. The single exception is Select (to change directories). All operations before the first Select are performed. The first Select is executed, and all other operations are discarded.

Unlike the Browse and Edit file selection panels, choosing Browse or Edit for multiple files does not place them all in the browse or edit ring. Each file is browsed or edited individually in the order in which it appears in the list.

If you choose copy or rename, the NAME field is an input field where you may type the new file name. When a copy operation is successful the original name re-appears in its original location, and the new file is inserted in the list in its proper sort order. When a rename operation is successful, the new name appears in the original location in the file list. If the new name is longer than the NAME field, leave the original name intact. When you press Enter, a Directory-Copy File or a Directory-Rename File panel appears

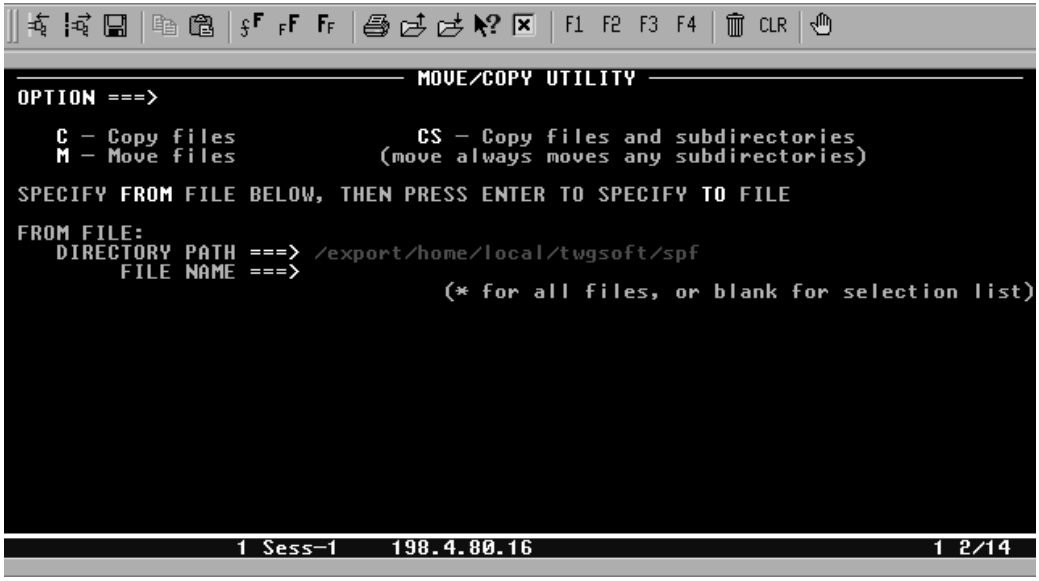


If you successfully delete a file, its name is replaced by *Deleted* in the file list.

You may also change file permissions from this panel. Move the cursor to the PERM'S field and type the desired file permissions. The change is effective immediately.

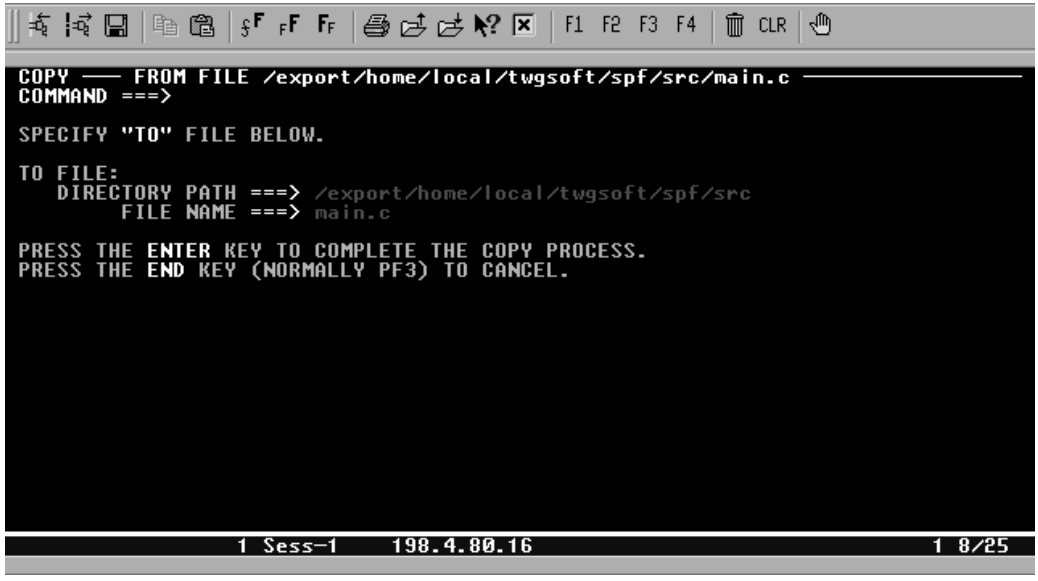
Move/Copy Utility (Option 3.3)

The Move/Copy Utility allows you to move or copy one or more files simultaneously. When you select 3 from the Utilities Menu panel, the Move/Copy panel appears. This is where you specify the source file(s) for the move or copy.

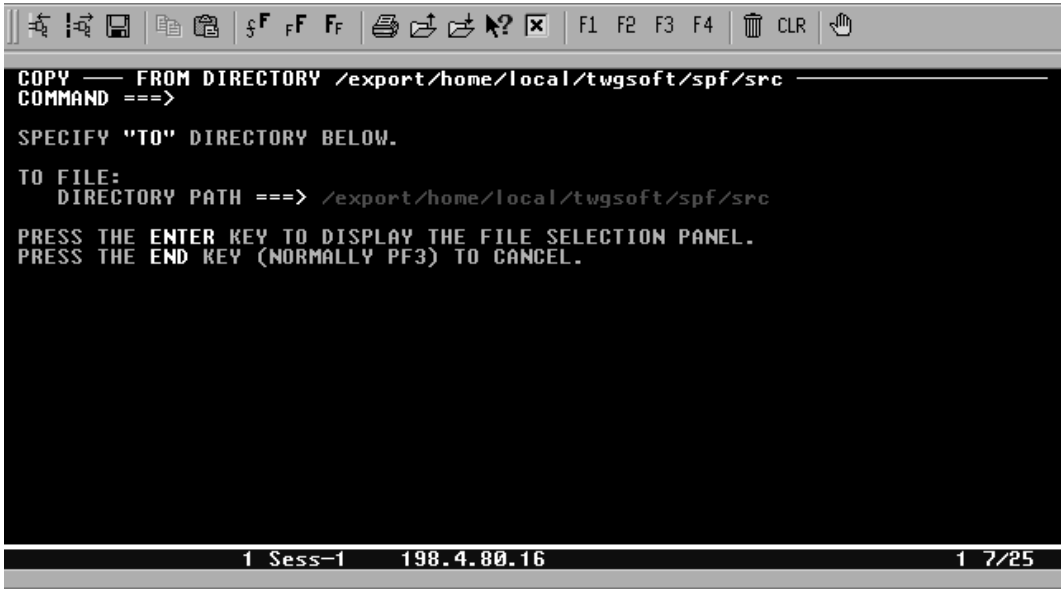


The Directory Path and File Name fields are automatically completed from the most recent use of these fields. You must specify Move or Copy in the primary input field.

The simplest use of this panel moves or copies a single file from one directory to another. Select a move or copy operation and make the desired modifications, if any, to the Directory Path and File Name fields. When you press Enter, a panel appears for you to specify the target directory and file name for the move or copy.



Another use of the Move/Copy utility is to move or copy several files from one location to another. Select a move or copy operation on the Move/Copy panel. Provide the source directory path and leave the file name blank. When you press Enter, a panel appears for you to specify the target directory for the move or copy.



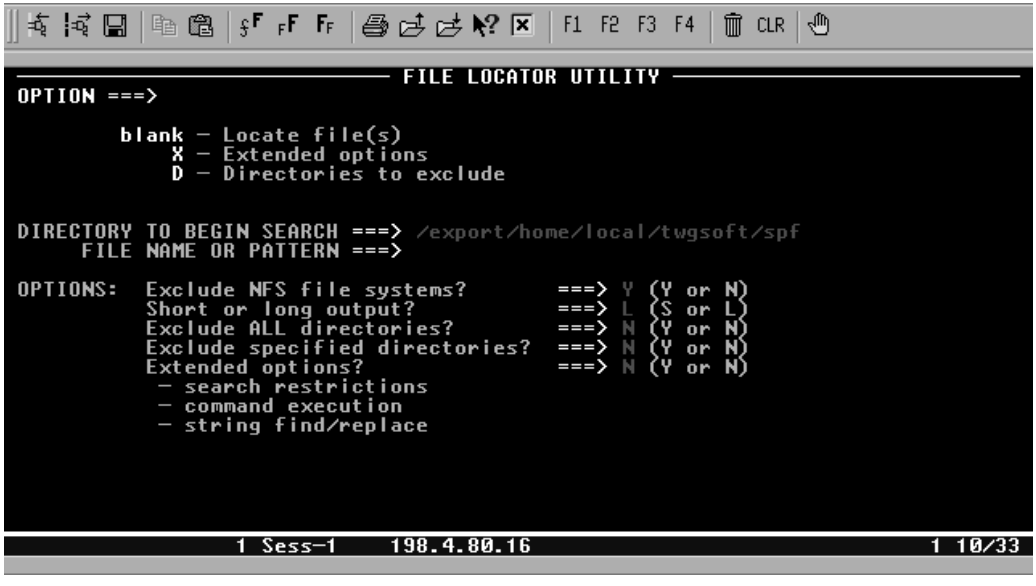
The directory path field is initially completed with the same directory path as the source directory. This facilitates specifying the target directory when it is a sub-directory of the source directory. Modify the target directory path as appropriate. When you press Enter, a file selection panel appears. Select one or more files to be moved or copied and press Enter.

An alternative method is available to move or copy multiple files with similar names. Select a move or copy operation on the Move/Copy panel. Provide the source directory path information. In the file name field, type a filename pattern using Unix wild-card characters. (***Chapter 2: Interaction with uni-SPF*** gives details on the use of wild-card characters.) Press Enter for the target directory panel and provide the appropriate directory path. When you press Enter from this panel, all files that match the filename pattern are automatically moved or copied.

File Locator Utility (Option 3.F)

The File Locator Utility searches directories for files that meet the specified criteria. It also provides options for automatically changing data within the files located. The results of the search are displayed in a file selection list where you may perform most of the operations available from the Directory Utility (Option 3.1).

When you choose Option 3.F, the following panel appears:



The fields on this panel allow you to specify the criteria for the current search.

DIRECTORY TO BEGIN SEARCH
: specifies the directory level at which this search begins. All subdirectories within this path are searched unless they are explicitly excluded using one of the options described below.

FILE NAME OR PATTERN
specifies the name of files to be located. This may be specified as a specific file name or as a file-name pattern that contains Unix wild-card characters. It may **not** be specified as a path name.

There are a number of options for qualifying the search. These include

Exclude NFS file systems?
Specify "Y" to restrict the search to local file systems. Specify "N" to permit search of NFS-mounted file systems. The default is "Y".

Use caution when allowing NFS searches. They may take a very long time if you have specified a starting directory that includes many NFS mount points.

Short or long output?
Specify "L" for output in the style of Option 3.1, including file name, permissions, owner, date/time stamp, size, and so forth. Specify "S" for output that is limited to the filename only. The default is "L".

Exclude ALL directories?
Specify "Y" to limit the search to only the directory specified. No subdirectories will be searched. Specify "N" to include subdirectories in the search. The default is "N".

Exclude specified directories?
Specify "Y" to exclude a specific list of directories. You must have previously defined this list using the "D" command, described below. Specify "N" to include subdirectories in the search. The default is "N".

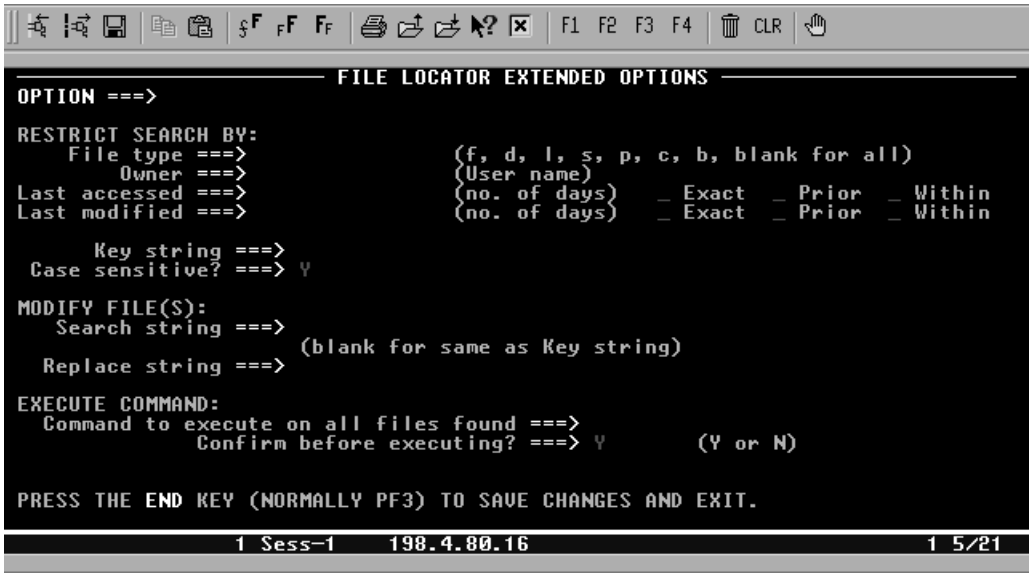
Extended options?
Specify “Y” to further qualify the search. You must have previously specified these options using the “X” command, described below. The default is “N”.

Use the “D” command in the primary input field to specify a list of directories to exclude from this search. You may specify up to 12 directories to be excluded. Specify only the simple directory name; do **not** specify directories as full path names. This rule applies even if the directory to be excluded is located one or more sub-directories below the starting directory.

The directories you specify for exclusion are saved as a “prune set” for use with future searches that begin at the same starting directory. Prune sets are saved in a table named ISPFLOC.tbl in each user’s HOME directory.

Set the “Exclude specified directories?” field to “Y” to use the prune set for this search. Set this field to “N” to ignore the prune set.

Use the “X” command in the primary input field to specify extended options that further qualify the search for files. The following panel appears:



The File Type, Owner, Last Accessed, Last Modified, and Key String fields limit the results of the search to the criteria specified.

For File Type, the types are

f	ordinary file
d	directory
l	symbolic link
s	socket
p	named pipe
c	character special device
b	block special device

Leave this field blank to find all types.

For Last Accessed or Last Modified, you must specify the number of days relative to today. Specify only one of Exact, Prior, or Within. The value specified must be a number. Exact finds only files with date stamps exactly ***n*** days ago. Prior finds only files with date stamps prior to ***n*** days ago. Within finds files with date stamps between ***n*** days ago and today.

The Key String field limits the results of the search to only those files that contain the specified string. The default search is case sensitive; specify “N” in the Case Sensitive field to perform a case-insensitive search.

Other fields on this panel allow you to perform operations on the files found before the file selection list appears.

Fields in the MODIFY FILES section allow you to modify data in the files found. Specify the search and replace strings in the fields provided. You may leave search string blank if you have specified a Key String and you want to change the key string in all files found. Strings specified for search-and-replace are used as specified – their case is preserved.

Fields in the EXECUTE COMMAND section allow you to specify a command to be executed on each file. By default, you must confirm the execution of the command for each file. Change the “Confirm before executing?” field to “N” to bypass this confirmation procedure.

Values set on the Extended Options panel are preserved so long as you remain within the File Locator Utility. Use the “Extended options?” field on the File Locator panel to control whether or not these options are used for a given search. When you exit the File Locator panel, the extended options are discarded; you must specify a new set each time you enter Option 3.F.

Job Status Utility (Option 3.J)

The Job Status Utility allows you to track and manage jobs submitted using the editor SUBMIT command or Option 5 (Background). Information about submitted jobs is maintained in the table ISPJOBS.tbl in the user's HOME directory. This table is updated each time a new job is submitted and each time the table is accessed or updated using Option 3.J.

When you choose Option 3.J, a panel similar to the following appears:

For background jobs, the PID of the job is captured immediately upon submission. Each time the table is accessed the status of that PID is checked. As long as the job is running, an "R" appears in the status field on the 3.J panel. Once the job is complete, an "*" appears in the status field.

For batch jobs, it is not possible to be certain when they will run. Access to their status depends on whether or not you access the table during the time the job is executing. The table is accessed each time you

submit a job and each time you enter the Option 3.J panel. If your job happens to be running at any of these times, the process-id is captured and its status is updated. If it has once been determined that a batch job is running, it can be determined when it completes.

That leaves open the possibility that a short-running job could get in, run, and get out between table accesses. Even for long-running jobs, the question of timing of table accesses relative to actual job execution means that the job may not ever appear in the status list. Also, some batch jobs may never get a status and it is up to the user to clean up the 3.J table when you know the job is done. The Unix batch processor sends email to you when the job is complete.

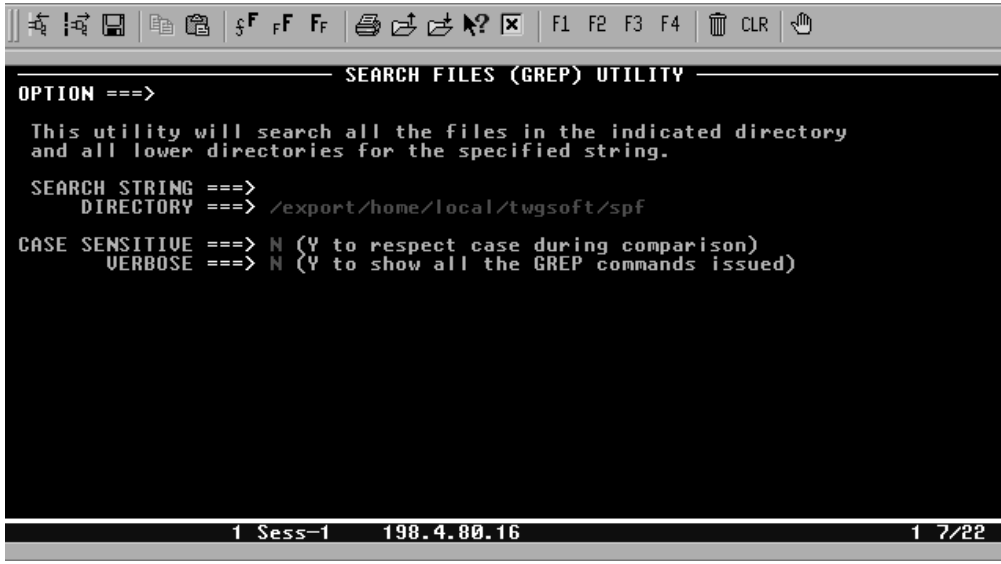
Besides viewing job status in 3.J, you may delete any record from the table. You may also cancel a pending or currently executing job from the 3.J panel.

Whenever you cancel a job, its files in .ISPsubmit are automatically removed. Whenever you delete a job record, its files in .ISPsubmit are automatically removed. For a batch job that does not show a PID, it is therefore important to be sure that the job has completed (or that it does not matter) before you delete its record.

Search Files Utility (Option 3.S)

The Search Files Utility searches for a given string using the Unix grep command. It searches all files in the directory specified and in all subdirectories therein.

When you choose Option 3.S, the following panel appears:



The Search Files Utility panel has the following fields:

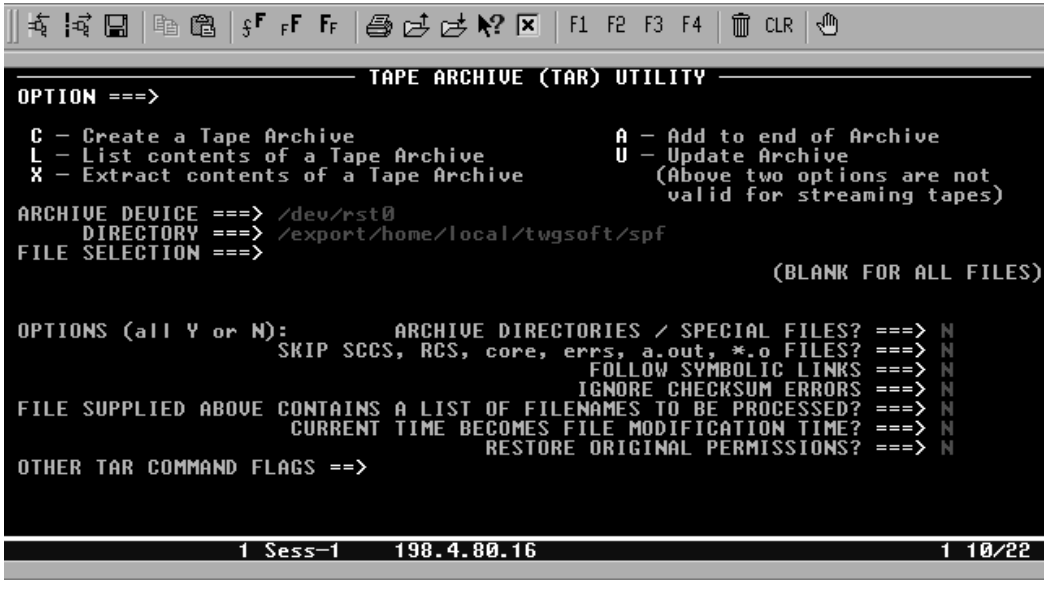
Field Name	Purpose	Type
SEARCH STRING	string to look for in all files	Required
DIRECTORY	top level of directory tree to search	Required
CASE SENSITIVE	should string search be case-sensitive; defauls is “N”	Optional
VERBOSE	search in verbose mode, in which each grep command executed is displayed along with the output; default is “N” (silent mode)	Optional

Search output is stored in a temporary file, with a file-name constructed from the process-id of this uni-SPF session. When the search is complete, this file is displayed in Browse. When you exit Browse, you may specify “K” to keep this file or press Enter to discard it. Note that your process-id remains the same for a sin-

gle uni-SPF session. If you wish to keep multiple files of search results from the same session, you must rename each file of search results before running the next search.

Tape Archive Utility (Option 3.T)

The Tape Archive Utility allows you to easily process tape archives. While these may most often be files on tape or diskette media, they may also be disk files. When you select T from the Utilities Menu panel, the



Tape Archive Utility panel appears.

You may

- create a tape archive
- list the contents of an archive
- extract files from an archive
- update an archive
- append an archive

Type the desired processing option in the primary input field.

The Archive Device field identifies the location of the tape archive to be processed. This is a device name for tape or diskette input or output or a full path name for disk file input or output.

The Directory and File Selection fields specify the source for the archive if you are creating, updating, or appending an archive. If you are extracting from an archive, the Directory field specifies a location for the output. If you are listing or extracting from an archive, the File Selection field limits the output. Leave the File Selection field blank to list or extract all files. Use Unix wild-card characters in the File Selection field to specify file name patterns. (C***hapter 2: Interaction with uni-SPF*** gives details on using wild-card characters.)

A number of processing options are available as indicated on the panel. The initial setting for each of these options is “N”. To use any of the processing options, change the “N” to “Y”.

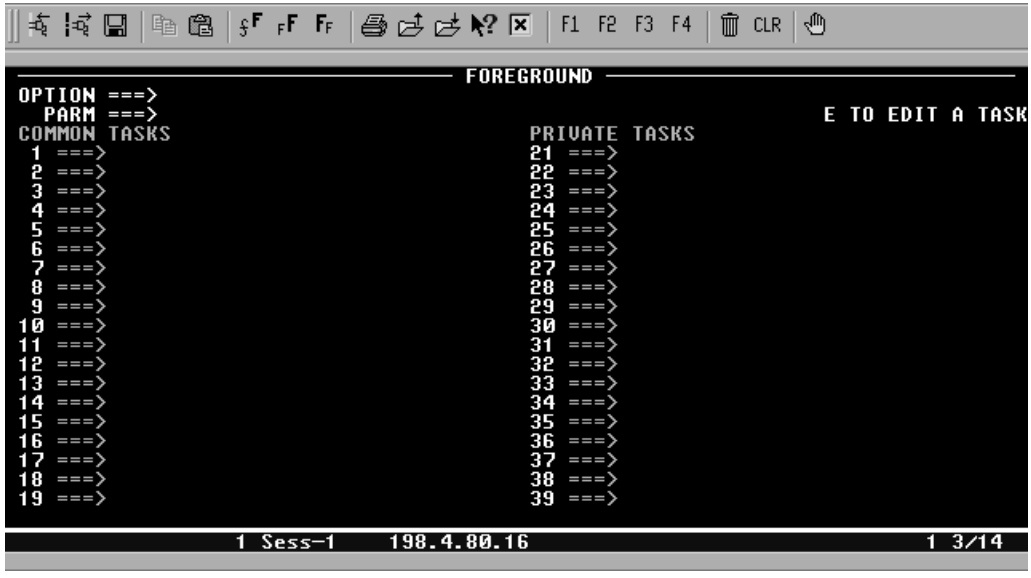
Chapter 7: Foreground

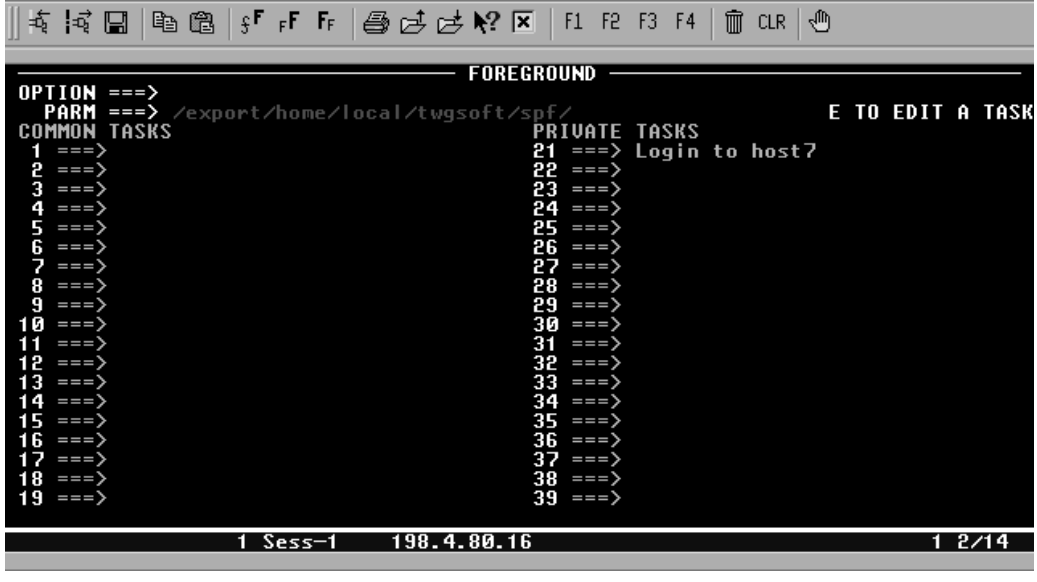
The Foreground facility allows you to define applications that can be executed as menu selections. You may define both Common Tasks and Private Tasks.

Common Tasks are shared by all users. Their definitions are stored in a common location, and all users see the same Common Task selections. To define a Common Task, you must have write permission for the directory where uni-SPF is installed. Thus, the installer or system administrator may be charged with defining and maintaining common tasks.

Private Tasks are unique to each user. Their definitions are stored in a table in the user's HOME directory (ISPFGNDP.tbl).

Examples of the Foreground panel appear on the following page. The first example shows this panel before any tasks have been defined. The second example shows how this panel might appear after some tasks have been defined. All users see the same options in the Common Tasks column. Each user sees different options in the Private Tasks column.


F1 F2 F3 F4 CLR
FOREGROUND
OPTION ===>
PARM ===>
E TO EDIT A TASK
COMMON TASKS
PRIVATE TASKS
1 ===>
2 ===>
3 ===>
4 ===>
5 ===>
6 ===>
7 ===>
8 ===>
9 ===>
10 ===>
11 ===>
12 ===>
13 ===>
14 ===>
15 ===>
16 ===>
17 ===>
18 ===>
19 ===>
21 ===>
22 ===>
23 ===>
24 ===>
25 ===>
26 ===>
27 ===>
28 ===>
29 ===>
30 ===>
31 ===>
32 ===>
33 ===>
34 ===>
35 ===>
36 ===>
37 ===>
38 ===>
39 ===>
1 Sess-1 198.4.80.16
1 3/14


F1 F2 F3 F4 CLR
FOREGROUND
OPTION ===>
PARM ===> /export/home/local/twgsoft/spf/
E TO EDIT A TASK
COMMON TASKS
PRIVATE TASKS
1 ===>
2 ===>
3 ===>
4 ===>
5 ===>
6 ===>
7 ===>
8 ===>
9 ===>
10 ===>
11 ===>
12 ===>
13 ===>
14 ===>
15 ===>
16 ===>
17 ===>
18 ===>
19 ===>
21 ===> Login to host7
22 ===>
23 ===>
24 ===>
25 ===>
26 ===>
27 ===>
28 ===>
29 ===>
30 ===>
31 ===>
32 ===>
33 ===>
34 ===>
35 ===>
36 ===>
37 ===>
38 ===>
39 ===>
1 Sess-1 198.4.80.16
1 2/14

The Parameter field automatically contains the directory path and file most recently used in an Edit or Browse session. This facilitates use of foreground applications that take a file name as the argument (compiling a source program is a typical example). You may change this information.

Executing Foreground Tasks

To run a Foreground application, type its number in the primary input field. The tab key moves automatically to the Parameter field if you need to modify this information for the application chosen. Press Enter to execute the application.

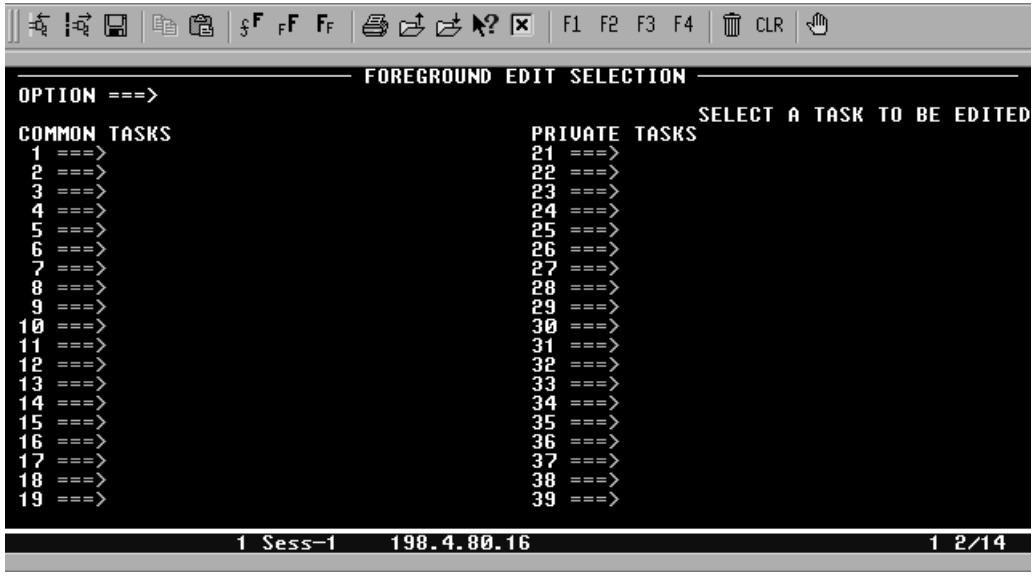
Output from a Foreground task is available for your review in one of two forms:

- If the task definition directs the output to a file, Foreground automatically invokes Browse to view that file when the task is complete. When you End Browse (normally PF3), the Foreground panel re-appears. This is the default.
- If the task definition directs the output to the screen, the Foreground panel scrolls up as output from the application appears. This is most commonly used for execution of programs that require terminal input. When you finish this interaction, you are prompted to “Press ENTER to continue uni-SPF”. The Foreground panel re-appears.

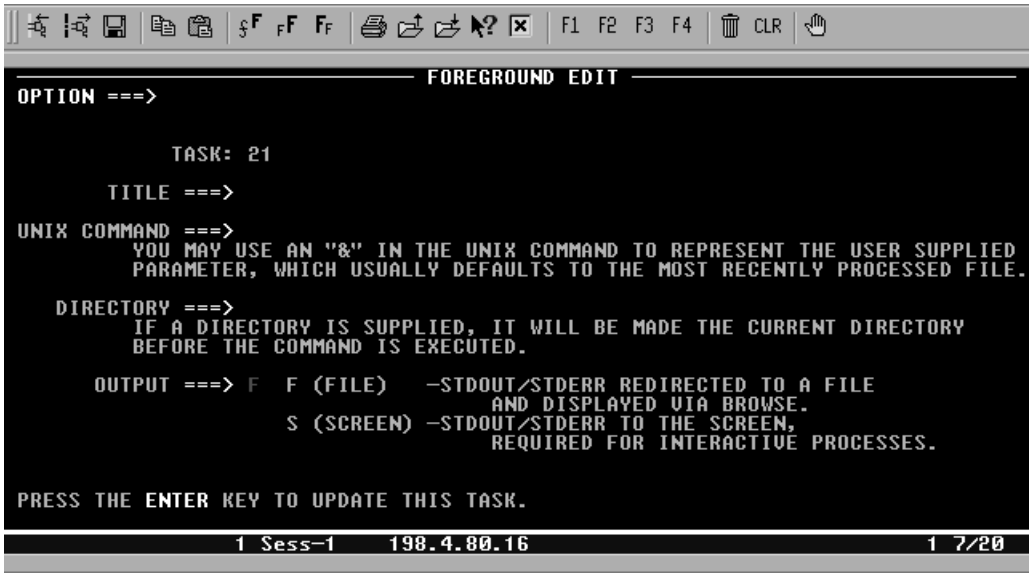
You may also run a Foreground task from any other panel in uni-SPF using the “jump” feature. Typing “=4.6" on any uni-SPF panel automatically executes Foreground task number 6. Subsequent displays are identical to what occurs when you run the application from the Foreground panel.

Defining Foreground Tasks

To define a foreground task or modify an existing definition, type “E” in the primary input field on the Foreground panel. The Foreground Edit Selection panel appears, prompting you to select the number of the task to be edited.



Type the number of the task you wish to define or modify in the primary input field. The Foreground Edit panel appears.



If you are defining a task for the first time, only the task number and the Output field contain information. If you are modifying an existing definition, the current definition is displayed.

In the Title field, provide the task description that you want to appear on the Foreground option panel. In the Unix Command field, provide the Unix command to be executed. You may use an “&” character in this command as a symbolic reference to a parameter to be provided when the task is executed.

The Directory field allows you to specify a directory in which this task is to be executed. If you complete this field, uni-SPF makes this the current directory before executing the foreground task. This is useful to insure that a program can access required files at execution time.

The Output field allows you to direct the output of the foreground task. By default, output goes to a file which is displayed in Browse for the user to review the

results of the task. This may be inappropriate for execution of programs that require terminal input, where interaction with the screen is required. For such tasks, type “S” in the output field.

Application definitions are stored in two locations:

- For common tasks, they are stored in a table named ISPFGNDS.tbl in the directory where the uni-SPF executable binary is installed.
- For private tasks, they are stored in a table named ISPFGNDP.tbl in each user’s HOME directory.

If you are attempting to define a common task and receive the message

```
    shared task table does not exist and you
cannot create it
```

this indicates that you do not have write permission to the directory where uni-SPF is installed.

To remove an individual task from the Foreground menu, edit that task and delete the contents of the Title, Unix Command, and Directory fields. You may remove all Private Tasks by deleting the file ISPFGNDP.tbl from your home directory. Similarly, you may remove all Common Tasks by deleting ISPFGNDS.tbl.

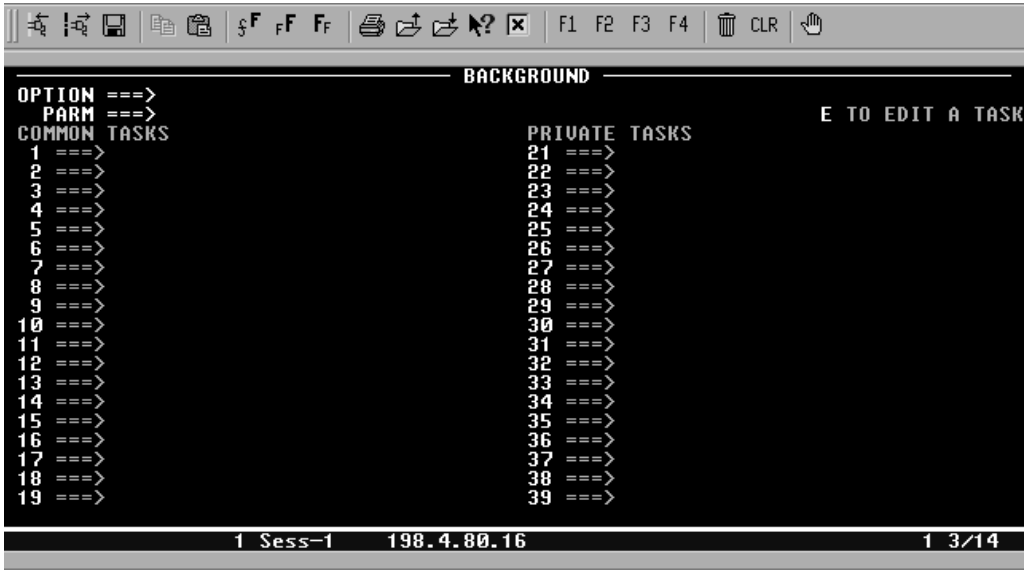
Chapter 8: Background

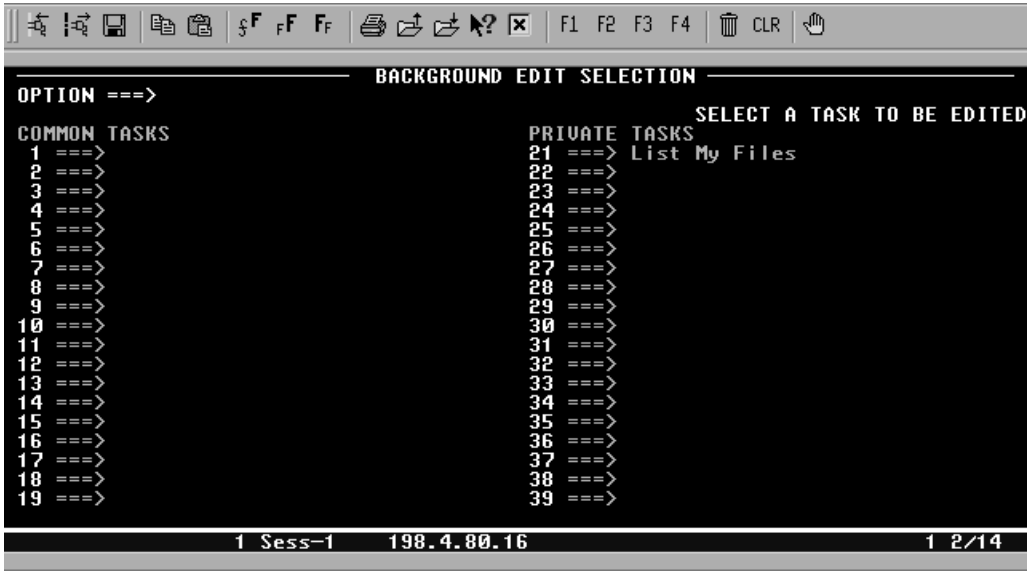
The Background facility allows you to define applications that can be executed as menu selections. You may define both Common Tasks and Private Tasks.

Common Tasks are shared by all users. Their definitions are stored in a common location, and all users see the same Common Task selections. To define a Common Task, you must have write permission for the directory where uni-SPF is installed. Thus, the installer or system administrator may be charged with defining and maintaining common tasks.

Private Tasks are unique to each user. Their definitions are stored in a table in the user's HOME directory (ISPBGNDP.tbl).

Examples of the Background panel appear on the following page. The first example shows this panel before any tasks have been defined. The second example shows how this panel might appear after some tasks have been defined. All users see the same options in the Common Tasks column. Each user sees different options in the Private Tasks column.


F1 F2 F3 F4 CLR
BACKGROUND
OPTION ===>
PARM ===>
E TO EDIT A TASK
COMMON TASKS
PRIVATE TASKS
1 ===>
2 ===>
3 ===>
4 ===>
5 ===>
6 ===>
7 ===>
8 ===>
9 ===>
10 ===>
11 ===>
12 ===>
13 ===>
14 ===>
15 ===>
16 ===>
17 ===>
18 ===>
19 ===>
21 ===>
22 ===>
23 ===>
24 ===>
25 ===>
26 ===>
27 ===>
28 ===>
29 ===>
30 ===>
31 ===>
32 ===>
33 ===>
34 ===>
35 ===>
36 ===>
37 ===>
38 ===>
39 ===>
1 Sess-1 198.4.80.16 1 3/14


F1 F2 F3 F4 CLR
BACKGROUND EDIT SELECTION
OPTION ===>
SELECT A TASK TO BE EDITED
COMMON TASKS
PRIVATE TASKS
1 ===>
2 ===>
3 ===>
4 ===>
5 ===>
6 ===>
7 ===>
8 ===>
9 ===>
10 ===>
11 ===>
12 ===>
13 ===>
14 ===>
15 ===>
16 ===>
17 ===>
18 ===>
19 ===>
21 ===> List My Files
22 ===>
23 ===>
24 ===>
25 ===>
26 ===>
27 ===>
28 ===>
29 ===>
30 ===>
31 ===>
32 ===>
33 ===>
34 ===>
35 ===>
36 ===>
37 ===>
38 ===>
39 ===>
1 Sess-1 198.4.80.16 1 2/14

The Parameter field automatically contains the directory path and file most recently used in an Edit or Browse session. This facilitates use of background applications that take a file name as the argument (compiling a source program is a typical example). You may change this field.

Executing Background Tasks

To run a Background application, type its number in the primary input field. The tab key moves automatically to the Parameter field if you need to modify this information for the application chosen. Press Enter to execute the application.

Output from a Background task is available for your review in one of two forms:

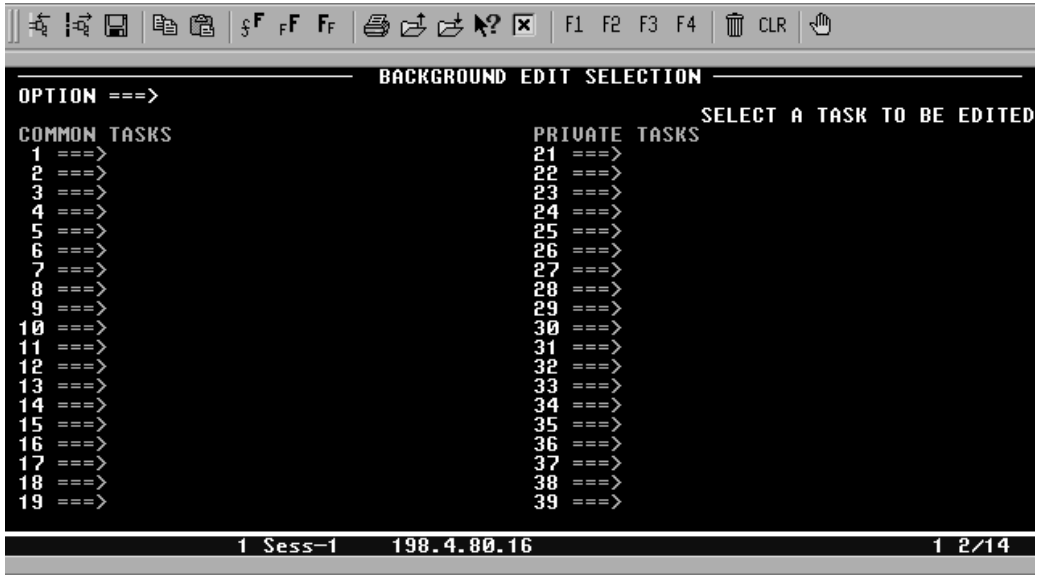
- If the task definition directs the output to a file, Background submits the task for Unix background execution and redirects the output to the file specified. You may then view the output through Edit or Browse at your convenience.
- If the task definition does not specify an output file, Background submits the task for Unix batch processing. The results of the batch job are returned to you as a mail message by your system's electronic mail facility.

You may also run a Background task from any other panel in uni-SPF using the "jump" feature. Typing "=5.6" on any uni-SPF panel automatically executes Background task number 6. Processing is identical to what occurs when you run the application from the Background panel.

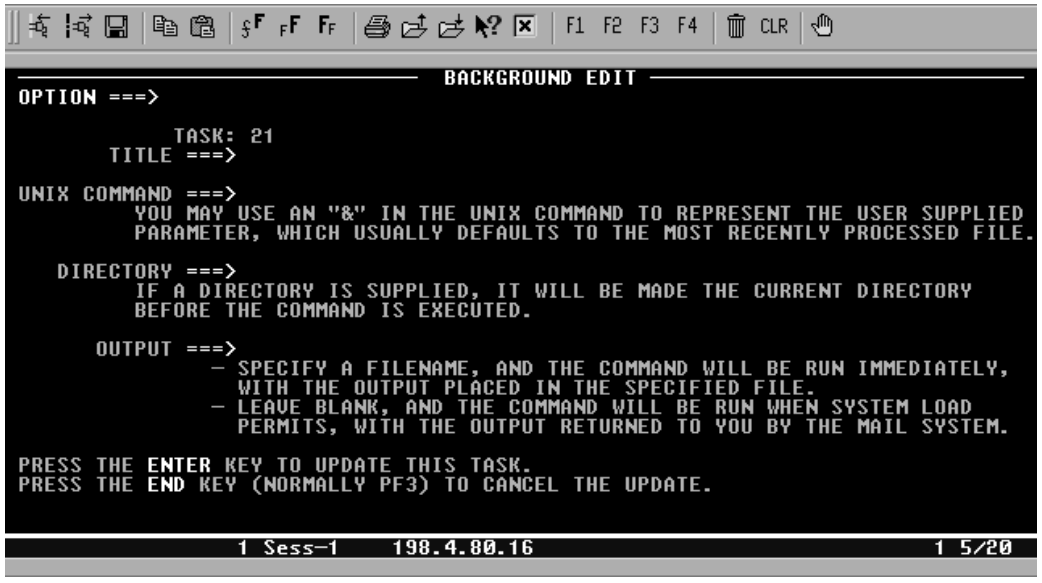
Background jobs submitted through Option 5 may be tracked using the Job Status Utility (Option 3.J). ***Chapter 6, Utilities*** describes the use of the Job Status Utility.

Defining Background Tasks

To define a background task or modify an existing definition, type “E” in the primary input field on the Background panel. The Background Edit Selection panel (shown below) appears, prompting you to select the number of the task to be edited.



Type the number of the task you wish to define or modify in the primary input field. The Background Edit panel appears.



If you are defining a task for the first time, only the task number field contains information. If you are modifying an existing definition, the current definition is displayed.

In the Title field, provide the task description that you want to appear on the Background option panel. In the Unix Command field, provide the Unix command to be executed. You may use an “&” character in this command as a symbolic reference to a parameter to be provided when the task is executed.

The Directory field allows you to specify a directory in which this task is to be executed. If you complete this field, uni-SPF makes this the current directory before executing the foreground task. This is useful to insure that a program can access required files at execution time.

The Output field allows you to direct the output of the background task. Specify a file name to redirect the results of the task to a file. Leave this field blank for

batch processing with the results returned to you by the mail system.

Application definitions are stored in two locations:

- For common tasks, they are stored in a table named ISPBGNDS.tbl in the directory where the uni-SPF executable binary is installed.
- For private tasks, they are stored in a table named ISPBGNDP.tbl in each user's HOME directory.

If you are attempting to define a common task and receive the message

```
shared task table does not exist and you cannot create it
```

this indicates that you do not have write permission to the directory where uni-SPF is installed.

To remove an individual task from the Background menu, edit that task and delete the contents of the Title, Unix Command, and Directory fields. You may remove all Private Tasks by deleting the file ISPBGNDP.tbl from your home directory. Similarly, you may remove all Common Tasks by deleting ISPBGNDS.tbl.

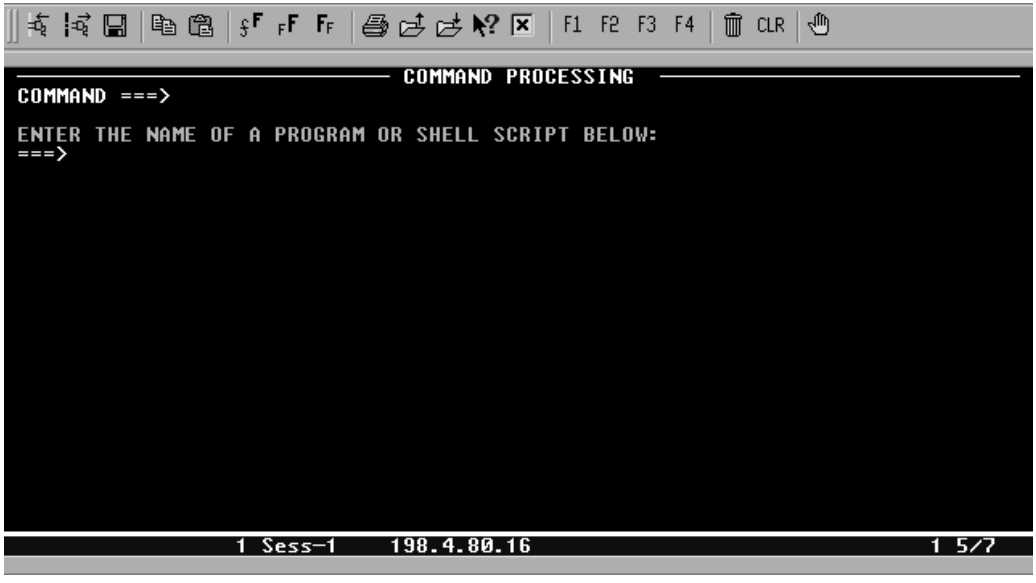
Chapter 9: Command

The Command facility allows you to perform Unix tasks without leaving uni-SPF. You may

- execute Unix commands
- run programs, shell scripts, or other command procedures
- login to another workstation on the network
- initiate background or batch tasks

or any other valid operating system activity.

When you select Command (Option 6) from the Main Menu panel, the Command Processing panel appears.

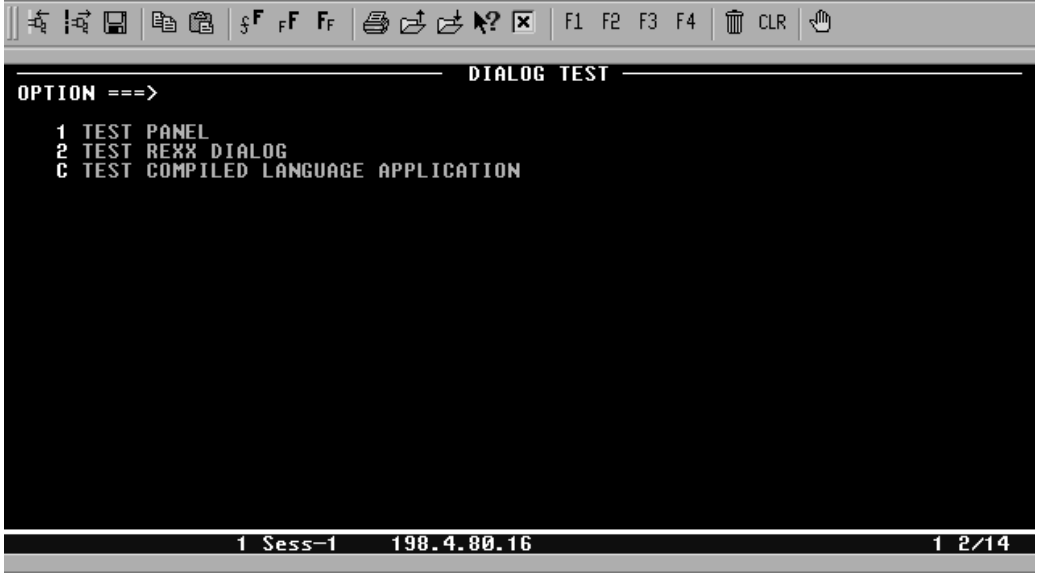


The cursor is automatically positioned at the input field where you enter the name of the command, program, or shell script to execute. Output appears at the bottom of the screen, followed by a prompt to "Press ENTER to continue uni-SPF". This allows you an opportunity to view your output before restoring the panel environment.

When you press Enter, the Command Processing panel appears. The command input field contains the most recently executed command.

Chapter 10: Dialog Test

The Dialog Test facility allows you to test dialog management programs and panel definition files without actually installing them in the system. When you select Dialog Test (Option 7) from the Main Menu panel, the Dialog Test panel appears.



Test Panel (Option 7.1)

When you select 1 to test a panel definition file, the Dialog Test-Panel screen appears.

Enter the name of the panel definition file. You may specify a full path name if the file is not in the current working directory. When you press enter, Dialog Test displays the panel defined in this file. The END command (normally PF3) returns you to the Dialog Test-Panel screen.

Test REXX Dialog (Option 7.2)

When you select 2 to test a dialog management program written in REXX, the Dialog Test-REXX Dialog screen appears. You may specify a full path name if the file is not in the current working directory.

When you press enter, the dialog management program is executed. When the dialog is complete, you return to the Dialog Test-REXX Dialog screen.

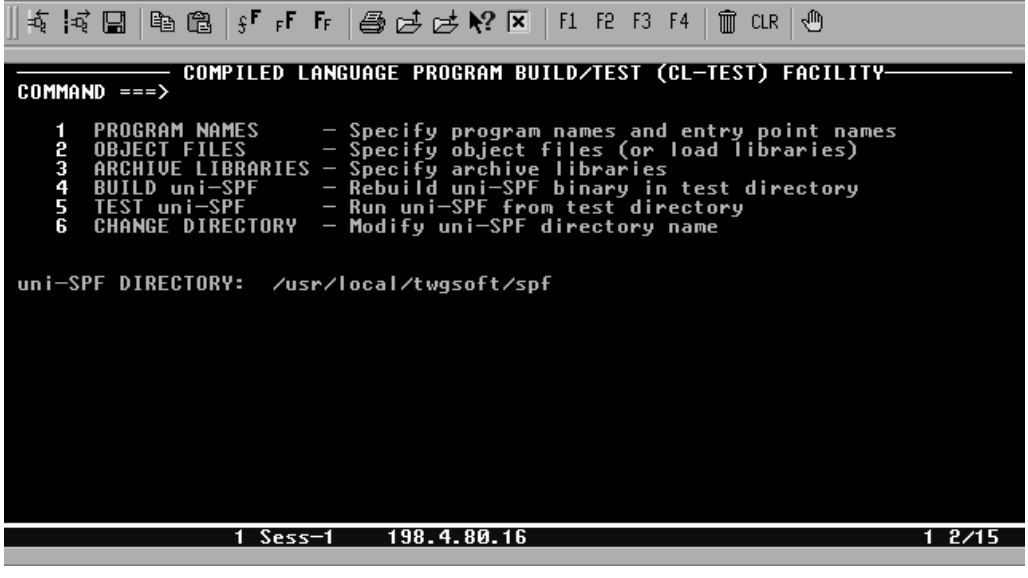
Test Compiled Language Application (Option 7.C)

The Test Compiled Language Application facility allows you to build and test Dialog Management applications written in a compiled language such as C, Fortran, PL/I, or COBOL. Details on the implementation of Dialog Management applications in compiled languages are provided in the ***uni-SPF Programmer's Reference Manual***.

This facility provides options to

- collect the information required to rebuild uni-SPF and store it in tables for future use
- rebuild the uni-SPF binary
- test the new binary using your compiled language programs

When you choose Option 7.C, the following panel appears:



The uni-SPF product directory name is determined automatically from the location of the binary that started the current session. If you wish to build and test your application in another location, you may change this field by selecting Option 6. The directory you specify must be a valid uni-SPF product directory structure with the following subdirectories:

bin, inc, lib, make, obj, src, and test

Choose Option 1 to specify the program names and entry point names for your application. This is required to establish the relationship between the name specified with the PGM keyword on the SELECT service and the entry point name in the compiled program. In many cases, these will be the same.

The first field (PGM-NAME) is the name used with the PGM keyword; the second field is the entry point in the program. You may modify this table at any time; however, you must update the API source file (src/spfTBL.c) using the "U" command before rebuilding uni-SPF if your changes are to be reflected in the new binary.

Choose Option 2 to specify the location(s) of your compiled object file(s). The first field is a short identifier of this application and is used both as a visual shorthand and as a key to sort the table. The object file name must be specified as a full path name to the ".o" file. You may modify this table at any time; however, you must update the make file (make/makefile.SPF) using the "U" command before rebuilding uni-SPF if your changes are to be reflected in the new binary.

Choose Option 3 to specify the names and locations of additional libraries required for this program. Programs written in Fortran, COBOL, or PL/I require their respective language libraries. The first field is the library name in the form "lib*xxxxxx*". The second field is optional and is the full pathname of the directory where

this library is located. If the named library is located in /lib or /usr/lib, it is not necessary to specify the directory name. For Solaris systems, if the directory is included in your current setting of the environment variable LD_LIBRARY_PATH, it is not necessary to specify a directory name. You may modify this table at any time; however, you must update the make script (make/make.SPF) using the “U” command before rebuilding uni-SPF if your changes are to be reflected in the new binary.

Choose Option 4 to rebuild the uni-SPF binary. Output from the build is captured in a file. When it is complete, you are automatically placed in “Browse” to review this file. The new binary is placed in the “test” directory with the name “uniSPF.std”.

Choose Option 5 to test the new binary with your program. Fields on this panel allow you to specify the name of the program to run, parameters to pass to it, and the location of panel, message, or table files required by the application.

Appendix A: System Dependencies

This appendix includes workstation- and terminal-specific information for platforms on which uni-SPF operates. If your particular workstation is not mentioned in this Appendix, there are no known system dependencies associated with it (or it is not supported for uni-SPF).

Hewlett Packard

The discussion which follows is pertinent to HP/9000 workstations running HP-UX. It applies specifically to the console keyboards for models in the 300 and 400 series. Some terminals used with models in the 700 and 800 series may also exhibit the behavior described.

HP-UX provides a fairly complete terminfo definition for HP-specific keyboards. However, the following problems exist.

There is no key available to assign to the reset function since all possible keys are already mapped through terminfo. You must use the generic key sequence CTL-R.

Backtab (SHIFT-TAB) is mapped incorrectly and therefore does not function properly. This can be remedied by using Option 0.k.1 to define a keybinding that assigns the sequence `\ei' to the keyname `BACKTAB'.

The terminfo definition for the F-keys is incorrect. Each function key transmits an additional carriage return that is not included in the terminfo definition.

Thus, each time you press a function key in uni-SPF, the effect is as if you pressed a PF key and then immediately pressed Enter.

The distribution media for uni-SPF contains two utility programs that re-program the function key hardware.

- **hpfkey** - reprograms the F-keys for compatibility with uni-SPF (do not send the extra carriage return)
- **hpfkeyr** - resets the F-keys to their original state

HP's X-based windowing system has a feature that allows you to display a function key menu at the bottom of each menu. The hpfkey program also assigns key names for this menu. The uni-SPF distribution media contains utility programs to enable and disable the menu display.

- **hpmenu** - enables the function key menu display
- **hpmenur** - disables the menu display

The ispf shell script automatically invokes hpfkey and hpfkeyr if the user's TERM environment variable is set to "hpterm".

Some HP console keyboards have only eight functions keys and there is no easy way to map other keys as F9-12. The four extra, unlabelled keys near the right of some keyboards simply mimic F5-8. Perhaps the best way to compensate for this deficiency is to redefine the first 8 PF keys to perform the most desirable functions. uni-SPF Option 0.3 allows you to modify PF key definitions. You would then use the generic key sequence for the PF key functions that are used less often.

The keybind library set contains the following keyboard mappings for the hpterm terminal type:

Key	Keyboard Function
Break	Reset
Shift-Tab	Backtab
CTL-X	Home

If your terminal type, as indicated in your TERM environment variable, is “hpterm”, uni-SPF will automatically import this keybind set. The section “Keyboard Mapping” in ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of keyboard mapping issues. ***Chapter 3: Parameters*** describes how to use the Keyboard Mappings option (Option 0.K) to define new mappings, modify existing ones, or import standard sets from the library.

Problems Using Function Keys. HP-UX does not support an internal timeout interval (the curses notimeout() function) when you login to the system across a network (rlogin, rsh, telnet). This limitation requires that keycodes to be translated by curses be received in a very short duration. Network delays often cause the characters sent by various keys (PF keys and arrow keys especially) to be delayed so that they do not meet the internal timing constraint and are not translated properly. This causes the characters sent by the key to appear on the screen at the location of the cursor when the key was pressed.

You may circumvent this problem by defining keybinds for all special keys on your keyboard. Use Option 0.K.2 for this purpose.

IBM Risc System 6000

The terminfo entries for the RS/6000 console keyboards (HFT terminal) provide most of the appropriate keyboard mappings for uni-SPF. The following are exceptions:

- the END key does not properly perform Erase-EOF
- backtab (SHIFT-TAB) does not function properly
- there is no reset key available

The keybind library set contains the following keyboard mappings for the aixterm terminal type:

Key	Keyboard Function
End	Erase-EOF
Shift-Tab	Backtab
Print Screen	Reset
Keypad Home	Home
Keypad Up	Up
Keypad PgUp	Up Page
Keypad Left	Left
Keypad Middle Key	Reset
Keypad Right	Right
Keypad End	Erase-EOF
Keypad Down	Down
Keypad PgDn	Down Page
Keypad Insert	Insert
Keypad Delete	Delete

If your terminal type, as indicated in your TERM environment variable, is “aixterm”, uni-SPF will automatically import this keybind set. The section “Keyboard Mapping” in ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of keyboard mapping issues. ***Chapter 3: Parameters*** describes how to use the Keyboard Mappings option (Option 0.K) to define new mappings, modify existing ones, or import standard sets from the library.

Problems Using Function Keys. The IBM RS/6000 does not support an internal timeout interval (the curses notimeout() function) when you login to the system across a network (rlogin, rsh, telnet). This limitation requires that keycodes to be translated by curses be received in a very short duration. Network delays often cause the characters sent by various keys (PF keys and arrow keys especially) to be delayed so that they do not meet the internal timing constraint and are not translated properly. This causes the characters sent by the key to appear on the screen at the location of the cursor when the key was pressed.

You may circumvent this problem in one of two ways:

- define keybinds for all special keys using Option 0.K.2
- use the special AIX environment variable ESCDELAY to control the internal timing interval

In AIX, the ESCDELAY environment variable controls the internal timeout interval for key code recognition. ESCDELAY specifies the timeout interval in units of 200 microseconds. The default value of ESCDELAY is 500, or 0.1 second. The setting required to circumvent the problem is dependent on the total network load. However, a value of 2000 (0.4 second) or 2500 (0.5 second) should be sufficient in most cases.

Sun Microsystems

The discussion that follows is pertinent to all workstations from Sun Microsystems. It applies to all versions of the Sun3 and the Sun4 (known variously as SLC, IPC, SPARCStation I, SPARCStation II). All system dependencies for Sun platforms are related to keyboard mappings. The discussion which follows pertains to the console keyboard only.

The uni-SPF distribution media includes the file spf.ttyswrc. This file contains the correct mappings to allow Open Look windows to properly recognize the F- and R-keys for type 2, 3, and 4 keyboards and the dF-keys for Type 5 keyboards. Copy this file to each user's HOME directory, renaming it .ttyswrc.

The .ttyswrc file is automatically executed when you create a new window. After you have located it in your HOME directory, you must start a new window for the mappings to be effective.

The keybind library set contains a standard keybind set for the Open Look command tool. It is valid for Type 2, 3, 4, and 5 keyboards for terminal type "sun-cmd". The F9 key is mapped because it is not present in the terminfo database. The F10-F12 keys are present only on the Type 4 keyboards but are not defined in the terminal database. For consistency among keyboards, the R1-R3 keys are mapped to PF10-PF12. Other R keys are mapped to other keyboard functions.

Keys	Keyboard Function
F9	PF9
F10 (Type 4)	PF10
F11 (Type 4)	PF11
F12 (Type 4)	PF12
R1	PF10
R2	PF10
R3	PF10
R4	Insert
R5	Delete
R6	Reset
R7	Home
R9	PF7
F11	Reset
R13	Erase-EOF
R15	PF8
Insert (Type 4)	Insert
Delete (Type 3 and 4)	Delete
Insert (Type 4 keypad)	Insert
Delete (Type 4 keypad)	Delete

If your terminal type, as indicated in your TERM environment variable, is sun-cmd, uni-SPF will automatically import this keybind set. The section "Keyboard Mapping" in ***Chapter 2: Interaction with uni-SPF*** contains a detailed description of keyboard mapping issues. ***Chapter 3: Parameters*** describes how to use the Keyboard Mappings option (Option 0.K) to define new mappings, modify existing ones, or import standard sets from the library.

Please note that there may be some incompatibilities between uni-SPF's requirements for .ttyswrc mappings and those of other applications. As examples

- **For the Open Windows system**

  In some types of Open Look windows, F1 is not available to emulate PF1 since Open Look uses F1 to display its own Help information. You can disable Open Look's usage of F1 by commenting out the following line in your .xinitrc file:

  ```
  xmodmap -e 'keysym F1 = Help'
  ```

  To comment out a line, add a "#" to the front so that it looks like

  ```
  # xmodmap -e 'keysym F1 = Help'
  ```

If you are using an Open Look Xterm window, the function keys (F1-F12) may not work on some systems because of a deficiency in the terminfo database entry. Use uni-KEY (Option 0.K.2) to define appropriate keybinds for these keys.

The cursor control arrow keys and function key F1 will not work properly with an Open Look PSTERM window. Use uni-KEY (Option 0.K.2) to define appropriate keybinds for these keys.

Index

C

D

E

F

S

T

U