SH20-2409-0

Program Product

Structured Programming Facility/Conversational Monitor System: Program Reference Manual

Program Number 5748-XT3

The Structured Programming Facility is a program development tool designed to take advantage of the characteristics of IBM 3270 display terminals, and to increase productivity in an interactive environment for users of both structured and conventional programming techniques.

The Structured Programming Facility/Conversational Monitor System is functionally equivalent to the Structured Programming Facility/Time Sharing Option program product, 5740-XT8, Version 2.2. The two products are fully compatible in display formats and operation, except for those features that are explicitly oriented to the Time Sharing Option or the Conversational Monitor System.

This manual provides detailed information on how to use this program product.

IBM

First Edition (September 1979)

This edition applies to Release 1, Modification Level 0, of the program product Structured Programming Facility/Conversational Monitor System (5748-XT3) and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult your System/370 Bibliography (GC20-0370) for the editions that are applicable and current.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments has been provided at the back of this publication. If the form has been removed, address comments to IBM Corporation, Dept. 824, 1133 Westchester Avenue, White Plains, New York 10604. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This document is divided into two parts. The first part contains general information about the use and operation of SPF. It includes the following major sections:

INTRODUCTION
GENERAL DESCRIPTION
SYSTEM REQUIREMENTS

The second part contains detailed information on the operation of SPF. It includes the following major sections:

INVOCATION AND TERMINATION
MEMBER LIST COMMANDS
SPF PARMS (OPTION 0)
BROWSE (OPTION 1)
EDIT (OPTION 2)
UTILITIES (OPTION 3)
FOREGROUND (OPTION 4)
BATCH (OPTION 5)
CMS COMMAND (OPTION 6)
TUTORIAL (OPTION 7)

RELATED PUBLICATIONS

- SPF/CMS General Information Manual, GH20-2197 - Provides an overview and functional description of SPF.
- SPF/CMS Installation and Customization Guide, SH20-2410 - Provides detailed information on how to install and custom tailor SPF.
- SPF/CMS Quick Reference Summary (Card), GX20-2347 - Provides a summary of SPF options and command formats.
- SPF/CMS Logic Manual - Describes SPF internal logic, program structure, and data areas (available in December 1979).
- SPF/CMS Program Listings (Microfiche) - Compiler and Assembler listings of SPF programs (available in December 1979).

A plastic overlay, SX20-1976, is also available to assist in the use of SPF program function keys on IBM 3275 or 3277 terminals.

SPF COMMAND FORMATS

In this document, the following notation conventions are used to describe SPF command formats:

- Command verbs and keywords, which must be typed exactly as shown, are represented with upper-case characters.
- Substitutable operands are represented with lower-case characters.
- Optional parameters are enclosed in square brackets.
- A choice of parameters is indicated with slashes.
- Default parameters are underscored.

Example: CAPS [ON/OFF]

The operand is optional. Either ON or OFF may be typed. If the operand is omitted, ON is the default.

(This page intentionally left blank.)

CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

The Structured Programming Facility (SPF) is a program development tool designed to take advantage of the characteristics of IBM 3270 display terminals, and to increase productivity in an interactive environment for users of both structured and conventional programming techniques. SPF can be used either by an individual programmer, or by many programmers working together on a project. SPF features which increase programmer productivity and simplify operation include:

- Display presentations and menus which prompt the user, reduce keystrokes, and reduce the opportunity for error.
- Support for multi-level programming libraries, automatic collection of library activity statistics, and printing of library contents.
- Full screen, context editing which allows additions and changes to multiple lines in a single interaction.
- Simple one-character edit commands for inserting, deleting, duplicating, or rearranging lines of source data.
- Forward, backward, and sideways scrolling of source data or listings, plus the ability to locate information by character string or line number.
- Split screen, allowing two SPF functions to be performed independently on the same display terminal.
- Use of program function keys for frequently performed SPF operations and commands.
- Menu-driven utilities for specification, maintenance and current status display of libraries and files.
- Menu interface to standard language processors (compilers and assembler) for execution in the foreground or CMS batch machine.
- Document preparation support, including text editing features and a menu interface to the SCRIPT/VS Document Composition Facility.
- Hardcopy log summarizing significant user actions during the session.
- Online tutorial for instruction and reference -- especially valuable for the occasional or novice user of SPF.

The Structured Programming Facility/Conversational Monitor System (SPF/CMS) operates as a CMS application and is invoked simply by entering the command "SPF". It operates under VM/370 CMS Release 5 (and subsequent releases unless otherwise specified) with either the System Extensions or Basic System Extensions program product. It supports 24-line, 32-line, and 43-line IBM 3270 display stations, equipped with either 12 or 24 program function keys.

This manual provides detailed information on how to use SPF/CMS.

GENERAL DESCRIPTION

STRUCTURED PROGRAMMING FEATURES

Structured programming emphasizes the use of segmentation and indentation. A source program normally consists of a large number of relatively small segments. The segments are stored as separate members of a programming library. Within a segment, source statements are block-indented under each "IF-THEN-ELSE" or "DO-WHILE" to show the control structure. Features of SPF which are specifically oriented toward structured programming include:

- Programming library support. The SPF library structure facilitates the maintenance and tracking of multiple program segments at varying version/modification levels.

- Ease of segmentation changes. One segment can easily be split into multiple segments, or multiple segments can be merged into one.

- Ease of indentation changes. Single statements or blocks of statements can easily be shifted left or right by a specified number of column positions.

- Insert in context. A "DO-END" pair, for example, may be coded on consecutive lines, and then space can be opened between the two lines to allow insertion of a block of code.

- Visual verification aids. A block of code may be temporarily excluded from display so that the space which it occupies on the screen is closed up. This facilitates visual verification of the control structure, particularly when the length of a segment exceeds the screen size.

DISPLAY FORMAT

SPF uses four basic types of display presentations.

1. Option Selection Menus - The user selects from a list of options by keying a one-character selection code and pressing the ENTER key. Example: The SPF primary option menu (Figure 1). This is the first display that the user sees after invoking SPF.

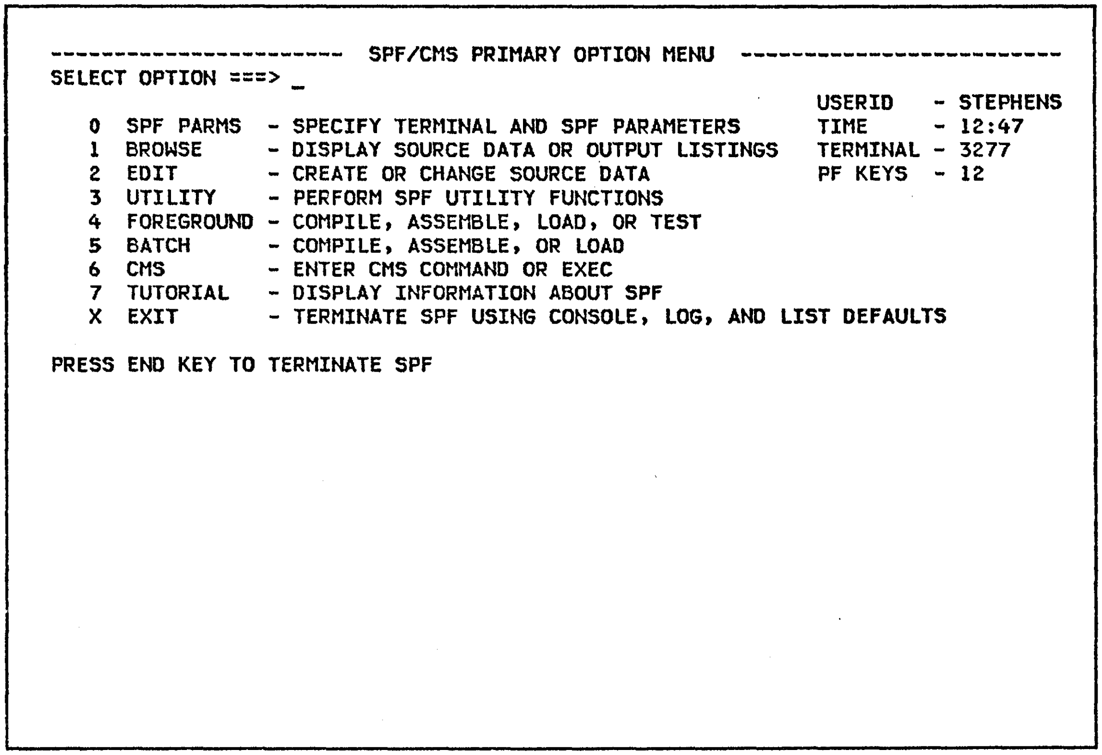
2. Parameter Entry Menus - The user supplies parameters by filling in labeled fields. In many cases, SPF pre-enters parameter values based on what the user last entered. Example: The browse file menu (Figure 2). This menu is displayed if the user selects option 1 from the primary option menu.

3. Member Selection Lists - Displays a list of members in a programming library. The user may select a member by entering a one-character code in front of the appropriate member name. Figure 3 shows an example of the member list on which the user has selected member COINS.

   The member selection list also shows the statistics for each member of an SPF library (see section entitled "Library Activity Statistics").

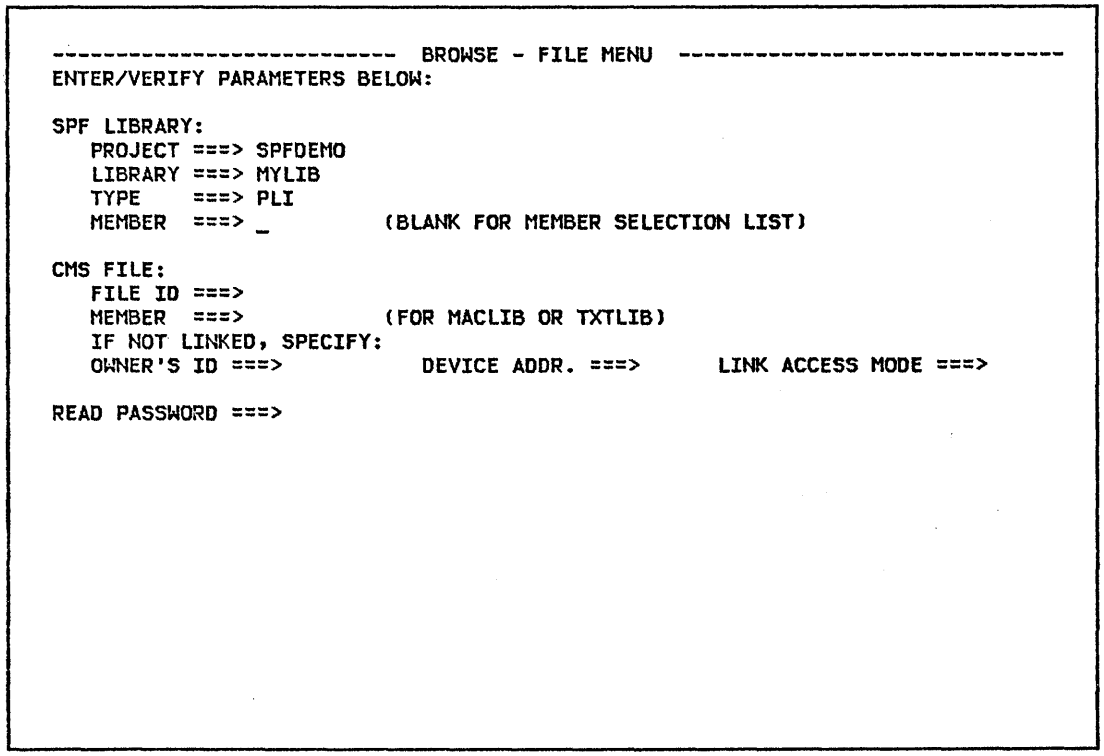
4. Data Display - for displaying source data or listings. Figure 4 shows an example of the browse display.

All SPF menus are formatted to fit on a 24-line screen. Displays which contain scrollable information, including member selection lists and data displays (browse and edit), occupy the full length of the screen (24, 32, or 43 lines).



```
----------------------- SPF/CMS PRIMARY OPTION MENU ------------------------
SELECT OPTION ===> _
                                                            USERID   - STEPHENS
   0  SPF PARMS  - SPECIFY TERMINAL AND SPF PARAMETERS      TIME     - 12:47
   1  BROWSE     - DISPLAY SOURCE DATA OR OUTPUT LISTINGS   TERMINAL - 3277
   2  EDIT       - CREATE OR CHANGE SOURCE DATA             PF KEYS  - 12
   3  UTILITY    - PERFORM SPF UTILITY FUNCTIONS
   4  FOREGROUND - COMPILE, ASSEMBLE, LOAD, OR TEST
   5  BATCH      - COMPILE, ASSEMBLE, OR LOAD
   6  CMS        - ENTER CMS COMMAND OR EXEC
   7  TUTORIAL   - DISPLAY INFORMATION ABOUT SPF
   X  EXIT       - TERMINATE SPF USING CONSOLE, LOG, AND LIST DEFAULTS

PRESS END KEY TO TERMINATE SPF
```

Figure 1. Primary option menu



```
--------------------------- BROWSE - FILE MENU -----------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB
   TYPE    ===> PLI
   MEMBER  ===> _          (BLANK FOR MEMBER SELECTION LIST)

CMS FILE:
   FILE ID ===>
   MEMBER  ===>            (FOR MACLIB OR TXTLIB)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>         DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>
```

Figure 2. Browse - file menu

SPF formats the first three lines of the display as follows:

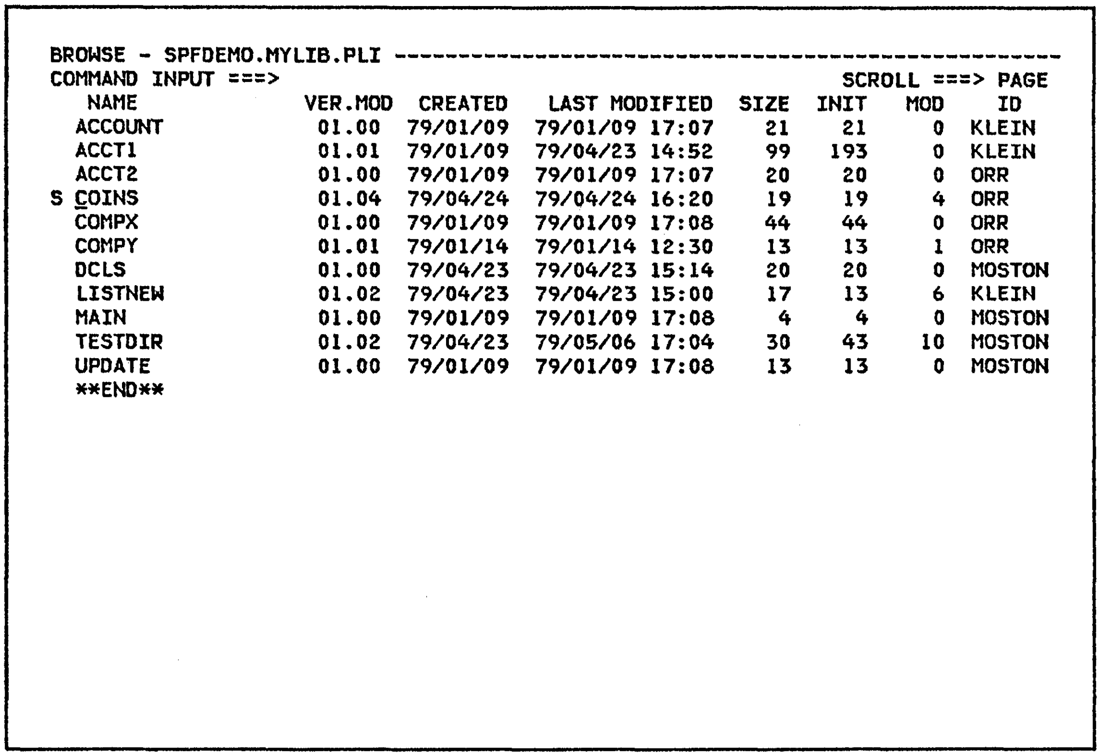
line 1	Title	Short Message
line 2	Prompt/Input	Scroll
line 3	Long Message	

The title area (line 1) identifies the function being performed and, where appropriate, the library or file identification, member name, version number and modification level. The short message area (line 1) is used to indicate:

- Current line number (browse) and column positions (browse and edit),
- Successful completion of a processing function,
- Error conditions (accompanied by audible alarm, if installed).

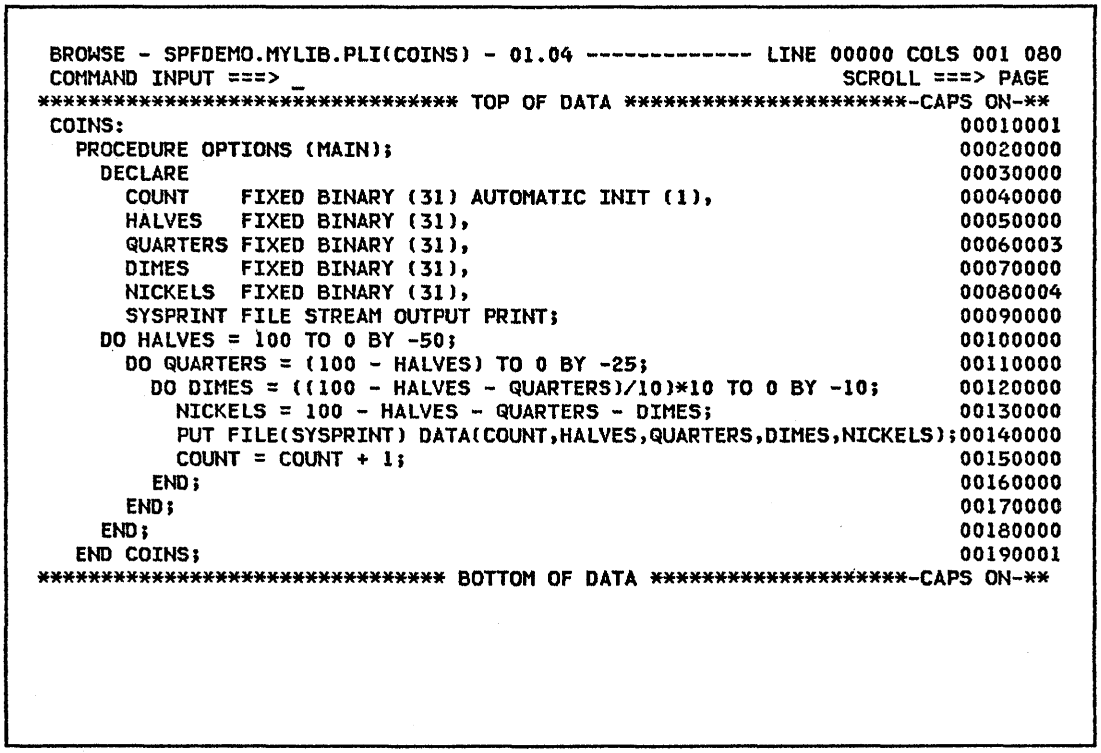
The prompt/input area (line 2) is used to enter an option selection or command. In cases where no option selection or command is applicable, this area contains a prompt. The scroll area (line 2) contains the current scroll amount whenever scrolling is applicable. The scroll amount may be overtyped by the user (see "Scrolling").

The long message area (line 3) is used to display an explanation of error conditions upon request (see "Help Information"). Normally this line is blank on menus, contains column headings on member selection lists, and is treated as part of the data area on data displays.



```
BROWSE - SPFDEMO.MYLIB.PLI ----------------------------------------------------
COMMAND INPUT ===>                                            SCROLL ===> PAGE
   NAME             VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT            01.00  79/01/09  79/01/09 17:07    21    21     0  KLEIN
  ACCT1              01.01  79/01/09  79/04/23 14:52    99   193     0  KLEIN
  ACCT2              01.00  79/01/09  79/01/09 17:07    20    20     0  ORR
S COINS              01.04  79/04/24  79/04/24 16:20    19    19     4  ORR
  COMPX              01.00  79/01/09  79/01/09 17:08    44    44     0  ORR
  COMPY              01.01  79/01/14  79/01/14 12:30    13    13     1  ORR
  DCLS               01.00  79/04/23  79/04/23 15:14    20    20     0  MOSTON
  LISTNEW            01.02  79/04/23  79/04/23 15:00    17    13     6  KLEIN
  MAIN               01.00  79/01/09  79/01/09 17:08     4     4     0  MOSTON
  TESTDIR            01.02  79/04/23  79/05/06 17:04    30    43    10  MOSTON
  UPDATE             01.00  79/01/09  79/01/09 17:08    13    13     0  MOSTON
  **END**
```

Figure 3. Browse - member selection list



```
BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------- LINE 00000 COLS 001 080
COMMAND INPUT ===> _                                          SCROLL ===> PAGE
********************************* TOP OF DATA **********************-CAPS ON-**
COINS:                                                                 00010001
  PROCEDURE OPTIONS (MAIN);                                            00020000
    DECLARE                                                            00030000
      COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),                   00040000
      HALVES   FIXED BINARY (31),                                      00050000
      QUARTERS FIXED BINARY (31),                                      00060003
      DIMES    FIXED BINARY (31),                                      00070000
      NICKELS  FIXED BINARY (31),                                      00080004
      SYSPRINT FILE STREAM OUTPUT PRINT;                               00090000
    DO HALVES = 100 TO 0 BY -50;                                       00100000
      DO QUARTERS = (100 - HALVES) TO 0 BY -25;                        00110000
        DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;      00120000
          NICKELS = 100 - HALVES - QUARTERS - DIMES;                   00130000
          PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);00140000
          COUNT = COUNT + 1;                                           00150000
        END;                                                           00160000
      END;                                                             00170000
    END;                                                               00180000
  END COINS;                                                           00190001
******************************** BOTTOM OF DATA ********************-CAPS ON-**
```

Figure 4. Browse - data display

SPF LIBRARIES

An SPF library is a collection of code or data units, called members. Each library generally contains members with the same type of information. Figure 5 shows a sample set of five SPF libraries which contain Assembler source, COBOL source, TEXT (object) modules, test data, and SCRIPT documentation.

SPF libraries are identified by project name, library name, and type, where:

- "project name" is the common identifier for all libraries belonging to the same project.
- "library name" identifies the particular set of libraries, such as MASTER, TEST, or ALICE.
- "type" identifies the type of information in the library, such as ASSEMBLE, COBOL, TEXT (object modules), DATA, or SCRIPT. Except for TEXT, the SPF type qualifier need not conform to the standard CMS naming conventions.

Most projects will require a hierarchy of related libraries to maintain effective version control over the programming development process and to reduce contention in library usage. Figure 6 shows a sample three-level hierarchy for a project, consisting of a set of master libraries, a set of test libraries, and three sets of development (private) libraries identified by user id.

New members or members undergoing changes will generally reside in the development libraries. A test library may be used to accumulate members which have been unit tested and are ready for integration test. A master library contains fully tested members, which may correspond to a previously released version of the system. Typically, a higher degree of control is maintained over the higher level libraries than the development libraries.

SPF allows concatenation of up to four SPF libraries during source editing, compilation, assembly, or SCRIPT/VS processing (plus additional MACLIBs for compilations and assemblies). Generally, the lowest level library is concatenated ahead of the next higher level library, and so on, in bottom-to-top order. For the hierarchy shown in Figure 6, a typical concatenation sequence might be library LARRY, followed by TEST, followed by MASTER. The concatenation applies to libraries of the same type, and is restricted to libraries which belong to the same project.

The purpose of concatenation during editing is to provide downward copying of a member from a test or master library to a development library. The edited member is saved in the development library (the first library in the concatenation sequence), while the unchanged version remains in the test or master library. When the new version is fully tested, it may be promoted back to a higher level library by means of the SPF move/copy utility.

The purpose of concatentation during language processing is to facilitate inclusion of source segments via INCLUDE or COPY statements (or SCRIPT "imbed" controls), and to allow debugging of new or modified programs without altering the contents of the test or master libraries. The output from the compilation or assembly (object module) is stored in the lowest level TEXT library (the first library in the concatenation sequence).

SPF libraries are maintained internally as CMS files. Each SPF library may consist of a set of CMS sequential files, or it may be a MACLIB or TXTLIB (for TEXT libraries only). The particular organization is designated when the library is specified to SPF via the file utility (option 3.2).

For more information on SPF libraries, see Appendix B.

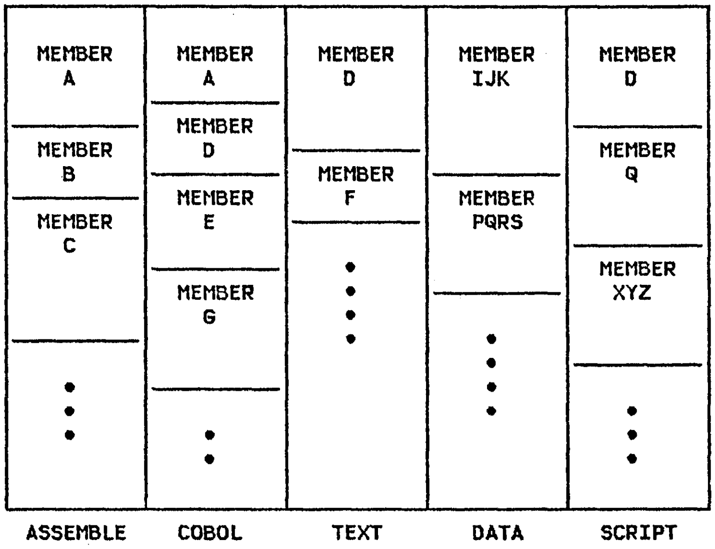


Figure 5. SPF library organization

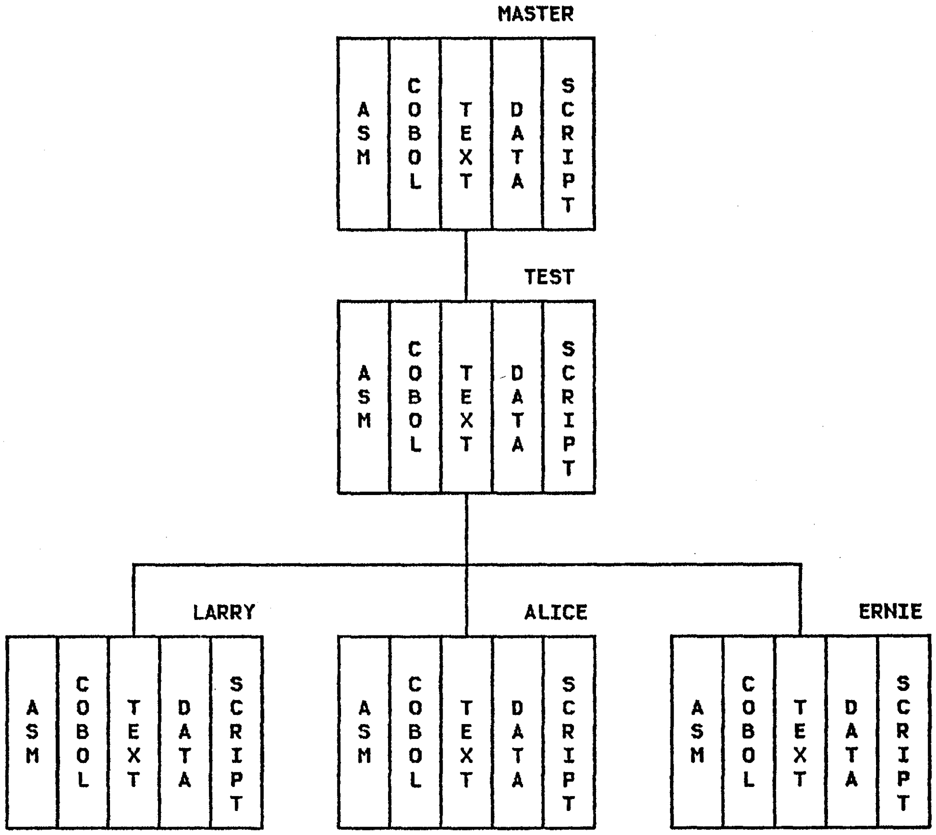


Figure 6. Hierarchy of SPF libraries

MENU PARAMETER SPECIFICATION

Several parameter entry menus require specification of an SPF library or CMS file identifier, and (when applicable) a read and/or update password. These are described in the following paragraphs.

Library and File Identification

To specify a member of an SPF library, the user must enter a project name, library name, type qualifier, and member name. Each of these may contain up to eight alphameric characters, of which the first character must be alphabetic. SPF automatically issues LINK and ACCESS commands, when required, to access the minidisk on which the SPF library resides.

SPF menus prompt the user for each component of the library identification as follows:

```
SPF LIBRARY:
   PROJECT ===>
   LIBRARY ===>
   TYPE    ===>
   MEMBER  ===>         (BLANK FOR MEMBER SELECTION LIST)
```

If the member name is not specified, a member selection list will be displayed.

When the library identification is displayed on a title line or message, SPF uses the following shorthand notation:

```
project.library.type(member)
```

That is, the project name, library name, and type qualifier are separated with periods, and the member name is enclosed in parentheses.

Some menus allow a concatenated sequence of up to four SPF libraries. For example:

```
SPF LIBRARY:
   PROJECT ===> SAMPLE
   LIBRARY ===> LARRY    ===> TEST     ===> MASTER    ===>
   TYPE    ===> COBOL
   MEMBER  ===>         (BLANK FOR MEMBER SELECTION LIST)
```

In this example, three COBOL libraries belonging to project SAMPLE would be concatenated in the following order of search:

```
SAMPLE.LARRY.COBOL
SAMPLE.TEST.COBOL
SAMPLE.MASTER.COBOL
```

SPF menus also allow users to specify other CMS files as follows:

```
CMS FILE:
   FILE ID ===>
   MEMBER  ===>          (FOR MACLIB OR TXTLIB)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>        DEVICE ADDR. ===>      LINK ACCESS MODE ===>
```

The CMS file id is entered according to normal CMS conventions, i.e.: filename, filetype, and (optionally) filemode, separated by one or more blanks.

If both an SPF library and a CMS file are specified on the same menu, the CMS file is used. In other words, to specify an SPF library, the CMS file id field on the menu must be blank.

If the CMS file is on a minidisk which is not currently linked and accessed, SPF will issue the appropriate LINK and ACCESS commands, based on user-specified LINK command parameters (owner's user id, virtual device number, and link access mode).

If a filemode is specified as part of the CMS file id, SPF will use the specified filemode letter in issuing the ACCESS command. Otherwise, SPF will use the first available (currently unused) filemode letter in issuing the ACCESS. Minidisks which are linked and accessed by SPF for the user will remain linked and accessed only as long as they are being used by a given function (e.g., edit).

Passwords

SPF provides a menu interface with the standard VM mechanism for virtual DASD read-only or read/write password protection as follows:

```
   READ PASSWORD ===>             UPDATE PASSWORD ===>
```

Non-display menu fields are used for both passwords.

If the minidisk containing the SPF library or CMS file is password protected, the user must enter one or both passwords. Only the read password need be supplied for read-only functions (e.g., browse). The update password must be supplied for functions which cause data to be written to the minidisk (e.g., edit). In edit, if concatenated SPF libraries are specified which reside on two or more minidisks, the first library is linked for update and the higher level libraries are linked read-only. Therefore both a read and update password may be required.

Note: The "update" password may require either the write (W) password or the multiple write (MW) password depending upon how the SPF library was specified (see utility 3.2) or, for a CMS file, depending on what was specified on the menu for link access mode.

Saving User Parameters

SPF retains default parameters and pre-enters appropriate information on menus according to what was last entered by the user. This includes the following types of information:

- Project name, library name(s), type, and password.
- Job statement information (for CMS batch submission).
- Defaults for console, log, and list processing.
- Terminal characteristics and PF key definitions.
- Edit profiles, including mask, tabs, and bounds.
- Processing options for each of the language processors.
- File specification/information parameters.

This information is automatically maintained in an SPF-controlled file (referred to as the user's SPFPARMS file) so that user information and defaults can be carried from one session to another. Accordingly, a new SPF user will have to enter each type of information the first time it appears on a menu. But from that point on, he simply verifies the information and makes whatever changes are desired before proceeding.

PROGRAM ACCESS AND FUNCTION KEYS

The program access (PA2), and 12 or 24 program function (PF) keys are used to request commonly used SPF operations. Keys which are not needed for SPF-defined key operations may be equated to edit and browse primary commands, or edit line commands, using the SPF parms option (option 0.3).

The default key arrangement for the 3-by-4 key pad (right-hand side of the keyboard) is shown in Figure 7. These are PF keys 1-12 on an IBM 3277 or 3275 terminal, or keys 13-24 on an IBM 3278 or 3276 terminal. A plastic overlay for the key pad, IBM order number SX20-1976, is available to assist with program key usage on 3277 or 3275 terminals.

For 24-key terminals, PF keys 1-12 have the same defaults as keys 13-24. It is strongly recommended that users of 24-key terminals continue to use the key pad (13-24) for SPF-defined operations, and redefine PF keys 1-12 to edit and browse commands.

PA1 ATTENTION	PA2 RESHOW

PF1 / 13 HELP	PF2 / 14 SPLIT	PF3 / 15 END
PF4 / 16 PRINT	PF5 / 17 FIND	PF6 / 18 CHANGE
PF7 / 19 ↑ (UP)	PF8 / 20 ↓ (DOWN)	PF9 / 21 SWAP
PF10 / 22 ← (LEFT)	PF11 / 23 → (RIGHT)	PF12 / 24 CURSOR

Figure 7. Default program key arrangement

The two PA keys are defined as follows. These may not be redefined by the user.

ATTENTION (PA1) Causes an immediate return to CP mode. This key should not be used, since it bypasses normal SPF termination. If it is pressed by mistake, a BEGIN command should be entered to resume SPF.

RESHOW (PA2) Redisplays the contents of the screen. May be useful if the ERASE or CLEAR key was pressed accidently, or if unwanted information has been keyed but ENTER (or a PF key) has not yet been pressed. Note that SPF does not support use of the Field Mark character (same key as PA2).

The SPF-defined PF key operations are described below. See Figure 7 for the default key assignments.

HELP Displays additional information about an error message or tutorial information about SPF commands and options.

SPLIT Causes split screen mode to be entered, or changes the location of the split (see "Split Screen").

END Terminates the current operation and returns to the previous menu. If the primary option menu is displayed, this key terminates SPF.

PRINT Causes a "snapshot" of the screen image to be recorded in the SPF list file.

FIND Repeats the action of the previous FIND command or the FIND part of the most recent CHANGE command (applies to browse and edit only).

CHANGE Repeats the action of the previous CHANGE command (applies to edit only).

UP Causes a scroll up (see "Scrolling").

DOWN Causes a scroll down (see "Scrolling").

SWAP Moves the cursor to wherever it was previously positioned on the other logical screen (see "Split Screen").

LEFT Causes a scroll left (see "Scrolling").

RIGHT Causes a scroll right (see "Scrolling").

CURSOR Moves the cursor to the first input field on line 2 (normally, the option selection or command input field). Pressing this PF key again causes the cursor to be moved to the second input field on line 2 (normally the scroll amount field).

RETURN Causes an immediate return to the primary option menu, bypassing any intermediate menus. (Logically equivalent to repeated use of the End key.) May also be used to go directly from one SPF option to another, without displaying the primary option menu, as follows: In any menu input field or primary command line, type an equal sign (=) followed by a primary option. Then press the Return PF key rather than the ENTER key.

PRINT-HI Same as PRINT except that high intensity characters on the screen are printed with overstrikes (using suppress-space carriage control) to simulate the dual intensity display.

NOP Causes the PF key to be functionless.

The RETURN, PRINT-HI, and NOP functions have no default key assignments. Users of 12-key terminals may wish to assign RETURN to an infrequently used PF key, such as PF4 (in lieu of PRINT) or PF12 (in lieu of CURSOR).

Caution on use of 3270 RESET key: On a 3277 or 3275, when the keyboard is locked during SPF processing, do not press RESET and attempt to enter additional information or use a PF key. Such action may produce unpredictable results.

SCROLLING

During edit and browse, the information to be displayed will generally exceed the screen size. Scrolling allows the screen "window" to be moved up, down, left, or right across the information. A member list can also be scrolled up and down, if it exceeds a single screen length.

Four program function (PF) keys are used for scrolling -- one for each direction. Whenever scrolling is allowed, a scroll amount is displayed at the top of the screen (line 2). This amount determines the number of lines (or columns) scrolled with each use of a Scroll PF key. The user may change the scroll amount by moving the cursor to the scroll field and overtyping the displayed amount. Valid scroll amounts are:

- A number from 1 to 9999 - specifies the number of lines (up or down) or columns (left or right) to be scrolled.

- PAGE - specifies scrolling by one page.

- HALF - specifies scrolling by a half page.

- MAX - specifies scrolling to the top, bottom, left margin, or right margin, depending upon which Scroll PF key is used.

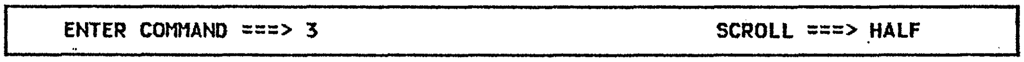
- CSR - specifies scrolling based on the current position of the cursor. The line or column indicated by the cursor is moved to the top, bottom, left margin, or right margin of the screen, depending upon which Scroll PF key is used. If the cursor is not in the body of the data, or if it is already positioned at the top, bottom, left margin, or right margin, a full page scroll will occur.

In edit, left and right scrolling is also affected by the current setting of the bounds. See description of BOUNDS command.

For scrolling purposes, a "page" is defined as the amount of information currently being viewed. On a 24-line display terminal, this is normally 22 lines by 80 columns for browse, or 22 lines by 72 columns for edit. (In split screen mode, the number of lines will be less than 22.) Thus, in full screen browse on a 24-line terminal, a scroll amount of HALF will move the window up or down by 11 lines, or right or left by 40 columns.

For member lists and browsing, the scroll amount is initialized to PAGE. For editing, the scroll amount is initialized to HALF. When the user overtypes the amount, the new value remains in effect until it is again changed by the user, or until a new member list or new data is displayed. The value MAX is an exception; following a MAX scroll, the scroll amount reverts to its previous value.

Any valid scroll amount can also be entered in the command input field and used in conjunction with a Scroll PF key. For example:



```
ENTER COMMAND ===> 3                                  SCROLL ===> HALF
```

If a Scroll PF key is pressed, the value in the command area will be used, overriding the normal scroll amount, without causing a change to the scroll field. This results in a one-time override; the value in the command area is blanked out after scrolling occurs.

If some key other than a Scroll PF key is pressed, the value in the command area will be interpreted as a command and will probably result in an error message.

OVERTYPING OF SCROLL AND MENU PARAMETERS

To reduce keystrokes, the scroll amount field can be changed by overtyping the first character(s) only:

- To change the scroll amount to PAGE, HALF, MAX, or CSR only the first character need be overtyped with "P", "H", "M", or "C" respectively.

- In the scroll amount field, any alphabetic characters following a number are ignored. For example "3AGE" is interpreted as "3".

The same rule applies to menu parameters in cases where the first letter uniquely defines the option. Examples:

```
REPLACE MEMBERS  ===> NO       (YES OR NO)
FILE DISPOSITION ===> OLD      (OLD OR MOD)
```

To change these options to YES or MOD, only the first character need be overtyped with "Y" or "M" respectively.

SPLIT SCREEN

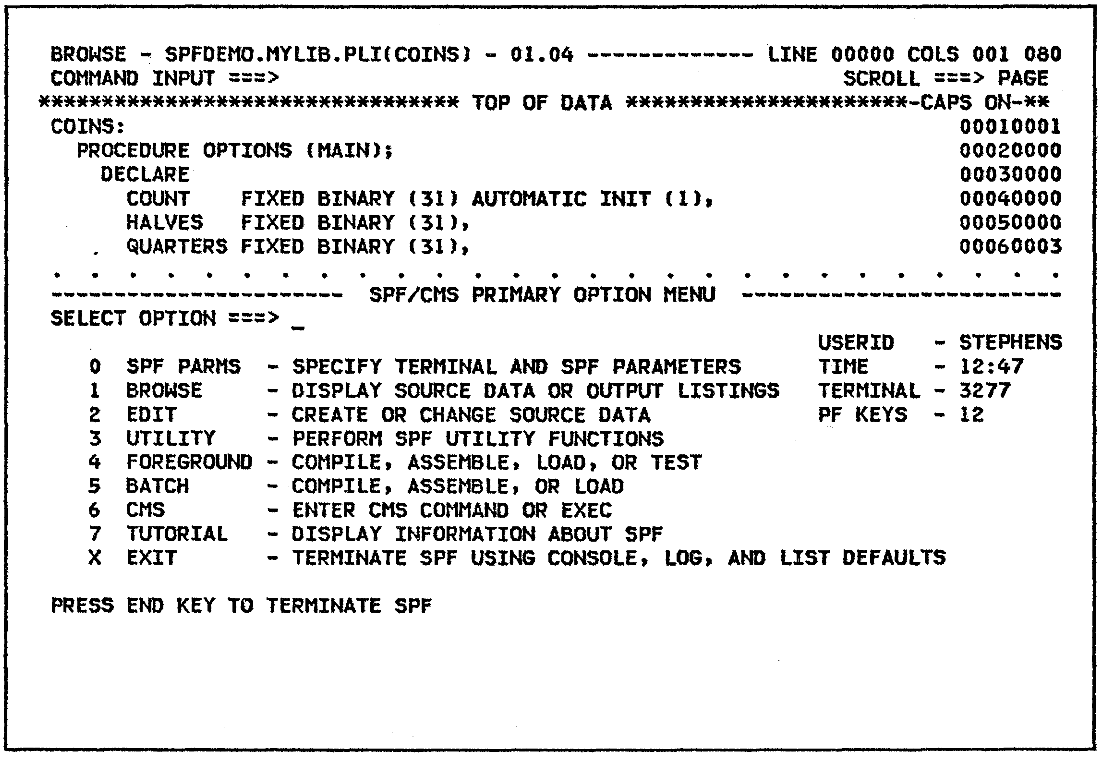
During edit and browse, the entire display screen is normally used for displaying data from one file. Sometimes it is helpful to view and/or edit two different (though probably related) SPF library members or files, or to be able to perform a utility function without leaving another SPF option. Split screen mode permits this dual use of the IBM 3270 terminal by splitting the screen into two "logical" SPF screens. The top and bottom screens are used independently to perform any of the functions provided by SPF.

The two logical screens are treated by SPF as though they were independent terminals. Functions that can be performed simultaneously on two separate terminals can be done on the top and bottom screens. Functions that are in conflict on two separate terminals (editing the same member of an SPF library or the same CMS sequential file, or browsing the same CMS file) will also be in conflict when simultaneously attempted on the top and bottom screens.

Split screen mode is entered by pressing the Split PF key, but first the user moves the cursor to the desired location for the split. The line containing the cursor will become the split line and will be identified by a row of periods. When the Split PF key is pressed, all lines below the cursor are treated as the bottom screen. The first display to appear on the bottom screen is the primary option menu. See Figure 8.

The split location may be repositioned at any time by moving the cursor to the desired line and pressing the Split PF key again. Repositioning may be required to bring an entire menu or tutorial page into view, since menus and tutorial pages are non-scrollable.

When SPF is being operated in split screen mode, one or the other of the logical screens is considered active at any point in time. Any interrupts, such as the operation of program function keys, are interpreted as having meaning for the active screen. The location of the cursor is used to identify which of the two screens is active. To switch from the top screen to the bottom screen, for example, move the cursor to the bottom screen. The Swap PF key may be used to move the cursor from one screen to another.



```
 BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------- LINE 00000 COLS 001 080
 COMMAND INPUT ===>                                             SCROLL ===> PAGE
********************************* TOP OF DATA ***********************-CAPS ON-**
 COINS:                                                                 00010001
   PROCEDURE OPTIONS (MAIN);                                            00020000
     DECLARE                                                            00030000
       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),                   00040000
       HALVES   FIXED BINARY (31),                                      00050000
     . QUARTERS FIXED BINARY (31),                                      00060003
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
 ----------------------- SPF/CMS PRIMARY OPTION MENU  -------------------------
 SELECT OPTION ===> _
                                                          USERID   - STEPHENS
    0  SPF PARMS  - SPECIFY TERMINAL AND SPF PARAMETERS   TIME     - 12:47
    1  BROWSE     - DISPLAY SOURCE DATA OR OUTPUT LISTINGS TERMINAL - 3277
    2  EDIT       - CREATE OR CHANGE SOURCE DATA          PF KEYS  - 12
    3  UTILITY    - PERFORM SPF UTILITY FUNCTIONS
    4  FOREGROUND - COMPILE, ASSEMBLE, LOAD, OR TEST
    5  BATCH      - COMPILE, ASSEMBLE, OR LOAD
    6  CMS        - ENTER CMS COMMAND OR EXEC
    7  TUTORIAL   - DISPLAY INFORMATION ABOUT SPF
    X  EXIT       - TERMINATE SPF USING CONSOLE, LOG, AND LIST DEFAULTS

 PRESS END KEY TO TERMINATE SPF
```

Figure 8. Split screen example

Note: Since only one logical screen is active at a time, the user cannot split the screen while executing a foreground compilation or a CMS command or EXEC under SPF.

In addition to moving the cursor to the other logical screen, the Swap PF key will automatically reposition the split if either logical screen is less than five lines long. For example, if the split occurs at the third line down from the top and the cursor is on the bottom screen, pressing the Swap PF key will move the cursor to the top screen and reposition the split to the third line up from the bottom.

Split screen mode is terminated by ending SPF processing on either logical screen (i.e., by pressing the End PF key when the primary option menu is displayed). The remaining logical screen is then expanded to its full size. Split screen may also be terminated by typing =X (exit option) in the command input area and pressing the Return PF key.

HELP INFORMATION

The "help" function allows the user to obtain additional information about any error message that has been displayed in the upper right-hand corner of the screen, or general information about an SPF command or option. If an error message is displayed, pressing the Help PF key causes a one-line explanation to be displayed. If this explanation is not sufficient, pressing the PF key again will display the appropriate section of the SPF tutorial for further information. If an error message was not displayed, the Help PF key causes the tutorial mode to be entered and displays the appropriate section of the tutorial. Once in tutorial mode, pressing the End PF key causes a return to the screen that was being viewed when the Help key was originally pressed.

LIBRARY ACTIVITY STATISTICS

Under user option, the SPF editor will automatically generate and maintain the following activity statistics for each member of an SPF library.

Version Number: Initialized to 1 when the member is created.

Modification Level: Number of times this version has been modified.

Creation Date: When this version was created.

Date/Time Modified: When this version was last modified.

Current No. Lines: Current size (number of records).

Initial No. Lines: Initial size of this version.

No. Modified Lines: Number of lines added or changed since this version was created. (Zero for unnumbered data.)

User Id: Who created or last updated this version (not the user id of the minidisk on which the library resides).

This information is maintained in a separate file (filetype SPFSTATS) on the same minidisk as the SPF library. The version number and/or user id may be changed via the "reset SPF statistics" utility (option 3.5).

For display and printing purposes, the following formats are used.

- For version number and modification level: VV.MM (e.g., "LEVEL 02.15" means version 2, modification 15).
- For creation date and date last modified: YY/MM/DD (e.g., "79/07/30" means July 30, 1979).
- For time last modified: HH:MM (e.g., "17:20" means 5:20 PM).

During browse and edit, the current version and modification level is displayed in the title area (line 1) following the library and member name.

The statistics are displayed next to each member name on member selection lists. The following abbreviations are used for column headings:

LIB	- library number (see below)
VER.MOD	- version number and modification level
CREATED	- creation date
LAST MODIFIED	- date and time last modified
SIZE	- current number of lines
INIT	- initial number of lines
MOD	- number modified lines
ID	- user id

The LIB column is displayed only if a concatenated sequence of SPF libraries was specified. It indicates the library (1, 2, 3, or 4) in which the member was found. For an example, see the description of Edit (option 2).

The statistics may be printed by requesting an "index listing" via the library utility (option 3.1). See Appendix C for a description of the index listing format.

LIST AND LOG FILES

Under user option, a listing of any source module that is created or modified by the SPF editor may be automatically recorded in an SPF listing file. Source listings and other types of printed output may also be obtained from the SPF utilities. A screen "snapshot" may be obtained by pressing the Print PF key.

SPF also maintains a log of significant user activities, including a record of files and SPF library members which were modified via edit or utility options, batch jobs which were submitted, etc. See Appendix C for a discussion of SPF listing and log formats.

The SPF list and log files are generated automatically, and reside on the user's A-disk during the SPF session. Their CMS file ids are:

```
SPFLIST LISTING
SPFLOG  LISTING
```

These files may be printed upon termination of SPF. A user option is available to prevent generation of the SPF log file (see SPF parms, option 0.2). Generation of the SPF list file may be avoided by simply not requesting any SPF print functions.

VIRTUAL CONSOLE

During SPF operation, all CMS commands and EXECs issued by SPF are spooled to the user's virtual console, including commands requested by the user (under options 4 and 6) and commands which are automatically executed on behalf of the user (such as LINK and ACCESS commands).

Upon termination of SPF, the user may specify if the virtual console is to be deleted or kept. If the user specifies that the console is to be deleted, SPF will purge the console and set it to "stop" status. If the user specifies that the console is to be kept, SPF will leave the console in "start" status. If the console remains in "start" status, it will be closed (released for printing) by CMS when the user logs off.

SYSTEM REQUIREMENTS

PROGRAMMING SYSTEMS

SPF operates as a CMS command under VM/370 CMS Release 5 (and subsequent releases unless otherwise specified). SPF is written in PL/S* and translated into Assembler Language. It uses the VM Communication Facility for inter-machine communication and uses the full-screen display support of either of the following program products, one of which must be installed:

VM/System Extensions program product (SEPP)	5748-XE1
VM/Basic System Extensions program product (BSEPP)	5748-XX8

* Programming Language/Systems (PL/S) is an IBM proprietary language.

SPF provides menu interfaces with the following IBM processing programs for foreground and batch execution:

VM/370 Assembler	
COBOL Compiler and Library	5740-CB1
COBOL Interactive Debug (foreground only)	5734-CB4
FORTRAN IV G1 Compiler	5734-F02
FORTRAN Interactive Debug (foreground only)	5734-F05
PL/I Checkout Compiler	5734-PL2
PL/I Optimizing Compiler	5734-PL1

The appropriate processing programs must be installed to use the SPF foreground and batch processing functions.

SPF also provides a menu interface to SCRIPT/VS to allow formatting, display, and printing of text maintained in SPF libraries or CMS files. Use of this feature requires installation of the following IBM program product:

Document Composition Facility (SCRIPT/VS) with the Foreground Environment Feature	5748-XX9

All the program-numbered items above can be ordered separately under IBM program product licensing agreements.

MACHINE CONFIGURATION

The computer system requirements are the same as needed for VM/370; that is, an IBM System/370 Model 135 and up with at least 384K of real storage, or an IBM 4300 series processor.

The storage requirements for the user's VM machine will vary depending upon the size of the files being edited or browsed, and the extent that "split-screen" is used. The minimum virtual machine size is 512K, but a larger size (at least one megabyte) is recommended. SPF will display warning messages if the amount of available virtual storage drops below 50K bytes. If all available storage is consumed an ABEND will occur, and the user must then re-IPL CMS.

TERMINALS

SPF supports the following IBM 3270 Display Stations:

3275 Model 2
3276 Models 2, 3, and 4
3277 Model 2 (local or remote attachment)
3278 Models 2, 3, and 4 (local or remote attachment)

The following keyboards are supported:

For 3275 or 3277 Display Stations:

78 Key Operator Console (feature 4632)
78 Key EBCDIC Typewriter (feature 4633)
78 Key EBCDIC Typewriter/APL (feature 4638), when operated with APL switch off.

For 3276 or 3278 Display Stations:

75 Key EBCDIC Typewriter (feature 4621)
87 Key EBCDIC Typewriter (feature 4627)
87 Key EBCDIC Typewriter/APL (feature 4626), when operated with APL switch off.
87 Key EBCDIC Typewriter/Text (feature 4629), when operated with TEXT switch off.

The standard character set (94 graphics plus blank and null) is supported on 3276 and 3278 Display Stations.

The following are supported, but not required:

Audible alarm (Feature #1090)
Print dual-case character set (RPQ #8K0366)

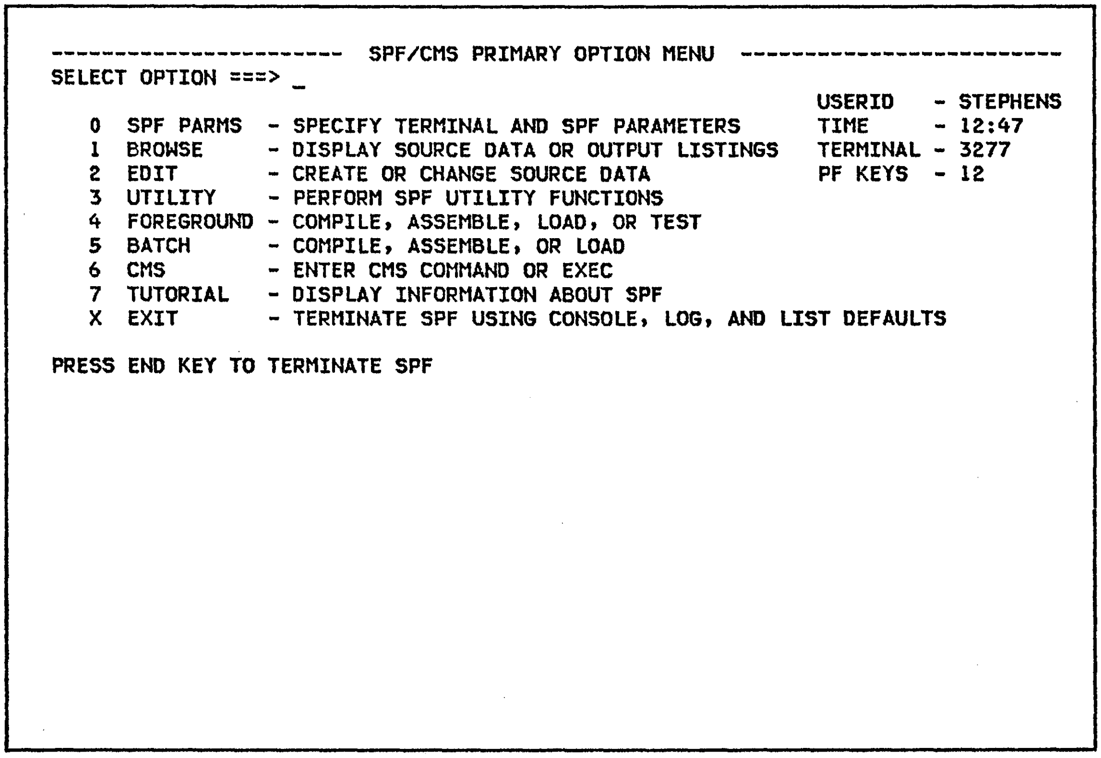
Installation of the audible alarm feature is strongly recommended to enhance usability. The alarm is sounded by SPF whenever a warning or error message is displayed.

INVOCATION AND TERMINATION

SPF is invoked by entering the command "SPF". No parameters are required. When SPF receives control it clears the screen, performs initialization functions, and displays the primary option menu (Figure 9). The user selects an option by typing a one-character code in the option field and pressing the ENTER key, e.g.

```
SELECT OPTION ===> 2
```

to select the edit option.



```
----------------------- SPF/CMS PRIMARY OPTION MENU ------------------------
SELECT OPTION ===> _
                                                          USERID   - STEPHENS
   0  SPF PARMS  - SPECIFY TERMINAL AND SPF PARAMETERS    TIME     - 12:47
   1  BROWSE     - DISPLAY SOURCE DATA OR OUTPUT LISTINGS TERMINAL - 3277
   2  EDIT       - CREATE OR CHANGE SOURCE DATA           PF KEYS  - 12
   3  UTILITY    - PERFORM SPF UTILITY FUNCTIONS
   4  FOREGROUND - COMPILE, ASSEMBLE, LOAD, OR TEST
   5  BATCH      - COMPILE, ASSEMBLE, OR LOAD
   6  CMS        - ENTER CMS COMMAND OR EXEC
   7  TUTORIAL   - DISPLAY INFORMATION ABOUT SPF
   X  EXIT       - TERMINATE SPF USING CONSOLE, LOG, AND LIST DEFAULTS

PRESS END KEY TO TERMINATE SPF
```

Figure 9. Primary option menu

The SPF primary options are:

- SPF PARMS - To specify SPF parameters and defaults, including: terminal type, number of PF keys, defaults for processing console, log, and list files, and PF key definitions.

- BROWSE - To display source data or output listings. Browse is intended primarily for viewing large files such as compiler listings or dumps.

- EDIT - To create or change source data, including program code, test data, documentation, etc. Unlike browse, edit reads the selected member or entire sequential file into virtual storage and retains it there during edit operations.

- UTILITIES - To print, rename, or delete SPF library members or CMS files, to specify SPF libraries, move or copy data between files, display or print SPF project listings, reset SPF library statistics, initiate spool output, retrieve data from the virtual reader, retrieve SPF libraries (via tape) from a TSO system, or format SCRIPT/VS documentation.

- FOREGROUND - To execute IBM language processing programs in the foreground, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), and COBOL or FORTRAN interactive debug. Also included is a menu interface to the LOAD command.

- BATCH - To generate and submit job statements and command streams for batch execution of IBM language processing programs, including: Assembler, COBOL, FORTRAN, and PL/I (checkout or optimizer); or for batch LOAD.

- CMS - To enter any CMS or CP command, or EXEC, during execution of SPF.

- TUTORIAL - To obtain immediate online instruction in the use of SPF. The tutorial may be viewed sequentially from beginning to end, or randomly by selecting topics from the table of contents or alphabetized index. The tutorial may also be entered from other SPF options by means of the Help PF key.

These options are described in the remainder of this manual.

To terminate SPF, the user must be out of split screen mode and must return to the primary option menu (via the End or Return PF key). Two termination options are available:

1. Option "X" may be entered on the primary option menu. This causes SPF to terminate with user defaults for processing console, log, and list files, as specified via SPF parms (option 0.2).

   Note: Option "X" may be used in conjunction with the Return PF key to immediately terminate split screen mode, or (in single screen mode) to immediately terminate SPF, as follows: In any menu input field or primary command line, type an equal sign (=) followed by an X. Then press the Return PF key.

2. The End PF key may be pressed to display the SPF termination menu (Figure 10) for specification of console, log, and list processing.

Upon termination of SPF, the screen is cleared, and the user is returned to normal CMS processing.

TERMINATION MENU

The termination menu allows the user to specify whether the virtual console is to be deleted (purged) or kept. If the user specifies that the console is to be deleted, SPF will purge the console and set it to "stop" status. If the user specifies that the console is to be kept, SPF will leave the console in "start" status.

The termination menu also allows the user to specify whether the log and list files are to be printed, deleted (purged), or kept without printing. If the files are to be printed, the following spool parameters may be specified:

- Number of copies - if more than one copy is desired.

- Spool class - for other than class A.

- For user - to spool the output to another user's virtual printer on the same VM system (ignored if either "to user/machine" or "remote node id" is specified).

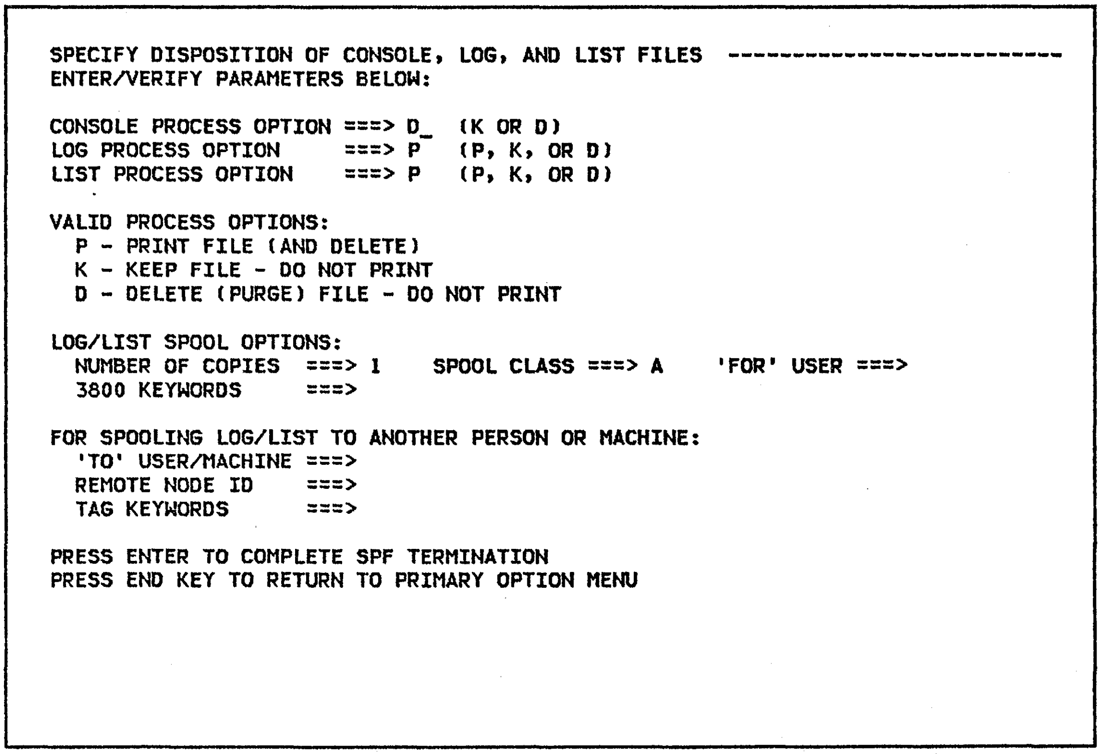
- 3800 keywords - if output is directed to an IBM 3800 printer which is attached to the CMS system as a virtual spooling device. See description of SPOOL command in the VM/370 CP Command Reference, GC20-1820.

- To user/machine - to spool the output to the virtual reader of another user or machine.
- Remote node id - if the desination is a remote node on the network (valid with either "to user/machine" or "tag keywords", but not both).
- Tag keywords - to specify control parameters for printing on a remote non-VM system. See description of TAG command in either the VM/370 Networking manual, SH20-1977, or the VM/370 RSCS Networking manual, SH24-5005.

Once the options have been entered on the termination menu, the ENTER key is pressed to complete SPF termination. Alternatively, the End PF key may be pressed to redisplay the primary option menu and continue SPF processing. (Use of the End PF key, in this instance, provides an escape mechanism in the event that the termination procedure was entered by mistake.)

Notes:

1. If the user selects option "X" but has not specified defaults via option 0.2, the termination menu is displayed.
2. The termination menu shown in Figure 10 is the version which is displayed if both log and list files have been used. If only one has been used during the SPF session, a different version of the menu is displayed showing the process option for either log or list, whichever is appropriate.



```
SPECIFY DISPOSITION OF CONSOLE, LOG, AND LIST FILES --------------------------
ENTER/VERIFY PARAMETERS BELOW:

CONSOLE PROCESS OPTION ===> D_  (K OR D)
LOG PROCESS OPTION     ===> P   (P, K, OR D)
LIST PROCESS OPTION    ===> P   (P, K, OR D)

VALID PROCESS OPTIONS:
  P - PRINT FILE (AND DELETE)
  K - KEEP FILE - DO NOT PRINT
  D - DELETE (PURGE) FILE - DO NOT PRINT

LOG/LIST SPOOL OPTIONS:
  NUMBER OF COPIES  ===> 1    SPOOL CLASS ===> A    'FOR' USER ===>
  3800 KEYWORDS     ===>

FOR SPOOLING LOG/LIST TO ANOTHER PERSON OR MACHINE:
  'TO' USER/MACHINE ===>
  REMOTE NODE ID    ===>
  TAG KEYWORDS      ===>

PRESS ENTER TO COMPLETE SPF TERMINATION
PRESS END KEY TO RETURN TO PRIMARY OPTION MENU
```

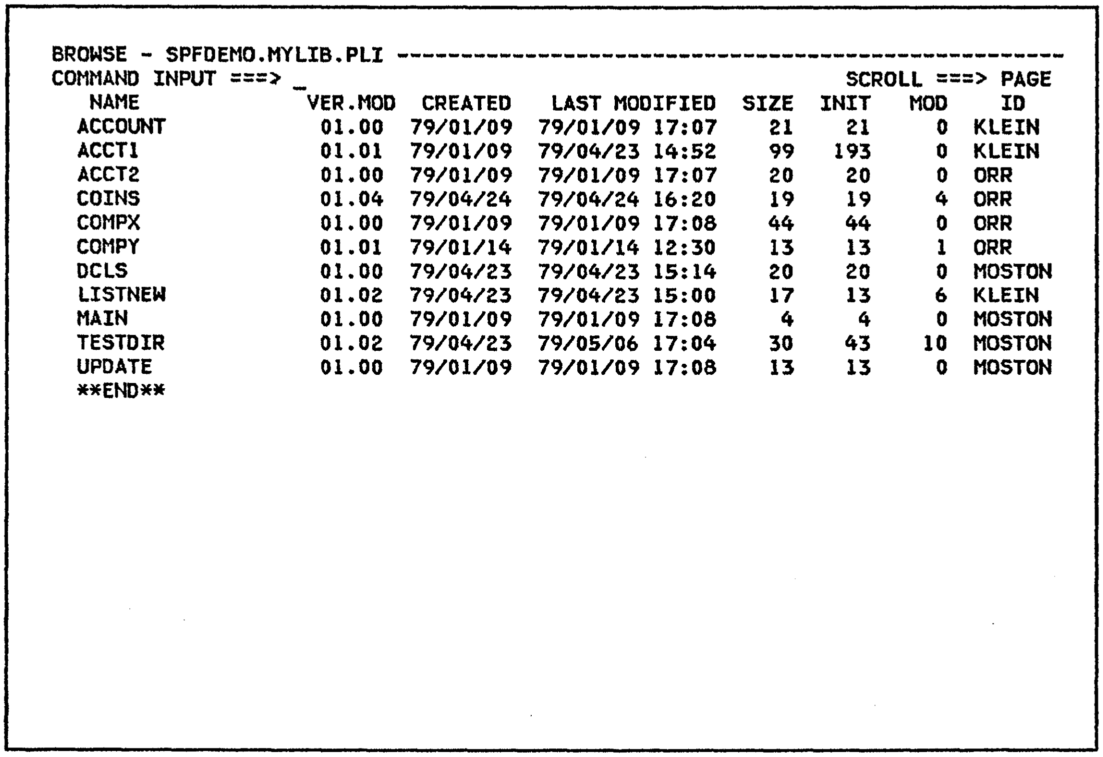
Figure 10. SPF termination menu

MEMBER LIST COMMANDS

For any of the following SPF options, a member selection list will be displayed if an SPF library or a CMS MACLIB or TXTLIB is specified but no member name is supplied:

- Browse
- Edit
- Library Utility
- Move/Copy Utility
- Reset SPF Statistics Utility
- SCRIPT/VS Utility

An example of a member list display is shown in Figure 11.



```
BROWSE - SPFDEMO.MYLIB.PLI ---------------------------------------------------
COMMAND INPUT ===> _                                     SCROLL ===> PAGE
   NAME             VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT           01.00  79/01/09  79/01/09 17:07    21    21     0  KLEIN
  ACCT1             01.01  79/01/09  79/04/23 14:52    99   193     0  KLEIN
  ACCT2             01.00  79/01/09  79/01/09 17:07    20    20     0  ORR
  COINS             01.04  79/04/24  79/04/24 16:20    19    19     4  ORR
  COMPX             01.00  79/01/09  79/01/09 17:08    44    44     0  ORR
  COMPY             01.01  79/01/14  79/01/14 12:30    13    13     1  ORR
  DCLS              01.00  79/04/23  79/04/23 15:14    20    20     0  MOSTON
  LISTNEW           01.02  79/04/23  79/04/23 15:00    17    13     6  KLEIN
  MAIN              01.00  79/01/09  79/01/09 17:08     4     4     0  MOSTON
  TESTDIR           01.02  79/04/23  79/05/06 17:04    30    43    10  MOSTON
  UPDATE            01.00  79/01/09  79/01/09 17:08    13    13     0  MOSTON
  **END**
```

Figure 11. Member list display

LOCATING MEMBERS

The member list may be scrolled up and down by means of the Scroll PF keys. In addition, a LOCATE command may be entered in the command input field on line 2 of any member list display. Format:

LOCATE member-name
Command abbreviations: LOC, L

This causes a direct scroll to the specified member name (i.e., the entry for the specified name will appear as the first line following the header lines). If the specified name is not found, scrolling is to the member name which (in the collating sequence) would immediately precede the specified name.

SINGLE SELECTION MEMBER LISTS

For browse, edit, and the SCRIPT/VS utility, one member at a time may be selected from the list. The member may be selected in either of two ways:

- Move the cursor down the left hand side of the screen and enter the single character "S" in front of the desired member name, or
- Enter a SELECT command in the command input field on line 2.

The format of the SELECT command is:

SELECT member-name
Command abbreviations: SEL, S

Use of this command allows selection of a member which is not in the list. For edit, selection of a non-existent member is the mechanism for creating a new member.

Upon termination of browse or edit (via the End PF key) the member list is redisplayed with an automatic scroll to the member just processed. The user may then select another member or press the End PF key again to return to the file menu.

Note: When the member list is redisplayed following browse or edit, it will not include new members which may have been created by another user (or, in split screen mode, on the other logical screen). To display an up-to-date list, return to the previous menu, leave the member name blank, and press ENTER.

MULTIPLE SELECTION MEMBER LISTS

For the other utility options, multiple members may be selected from the list. A single character code may be typed in front of one or more member name(s) before pressing the ENTER key. For the move/copy and reset utilities, the selection code is "S". For the library utility, the allowable codes are "P" (for print), "R" (for rename), "D" (for delete), and "B" (for browse).

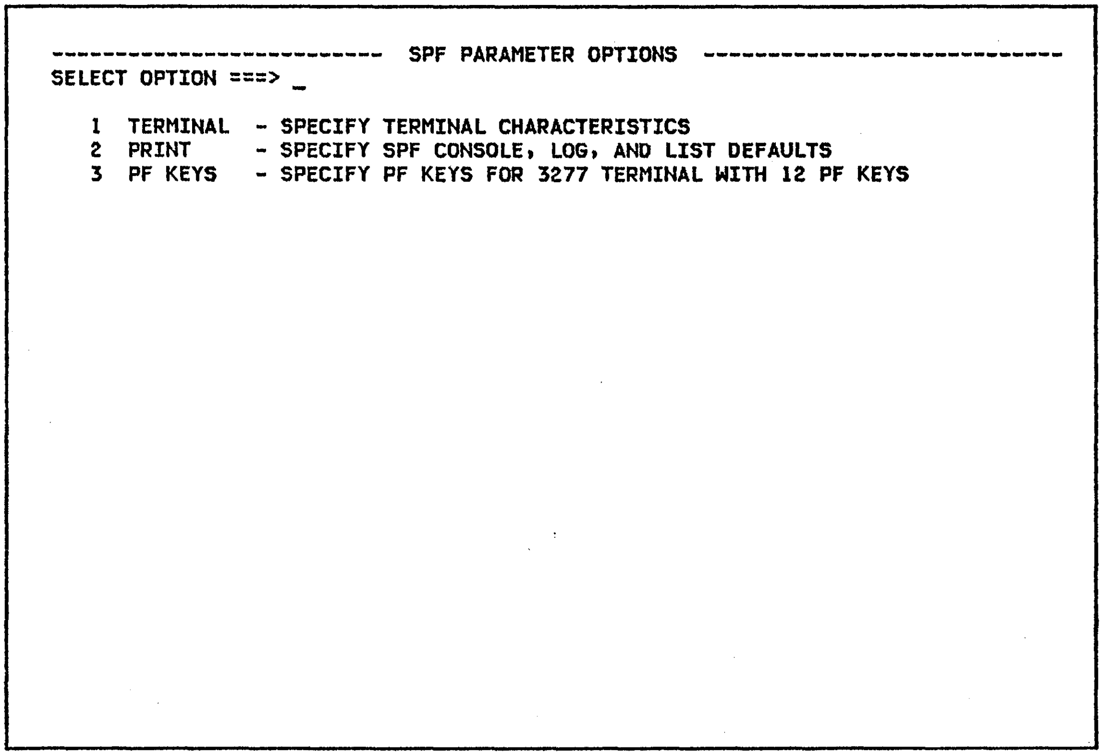
S	- Select member (except library utility)
P	- Print member (library utility only)
R	- Rename member (library utility only)
D	- Delete member (library utility only)
B	- Browse member (library utility only)

When ENTER is pressed, the selected member(s) are processed and a confirmation indicator is displayed to the right of each member name that was processed. See description of the utilities for examples. The user may then select additional members (scrolling to bring them in view, if necessary) or press the End PF key to return to the file menu.

SPF PARMS (OPTION 0)

The SPF parms option allows the user to display and change a variety of SPF parameters at any time during the SPF session. Changes remain in effect until the user changes the parameter again. Furthermore, these parameters are automatically retained for each user, so that current values remain in effect across SPF sessions.

When the SPF parms option is selected (primary option 0) a secondary selection menu is displayed (Figure 12). This menu may be bypassed by entering two numbers, separated by a decimal point, on the primary option menu. For example, entering "0.2" on the primary menu has the same effect as entering "0" on the primary menu and "2" on the secondary menu.



```
-------------------------- SPF PARAMETER OPTIONS ---------------------------
SELECT OPTION ===> _

   1  TERMINAL  - SPECIFY TERMINAL CHARACTERISTICS
   2  PRINT     - SPECIFY SPF CONSOLE, LOG, AND LIST DEFAULTS
   3  PF KEYS   - SPECIFY PF KEYS FOR 3277 TERMINAL WITH 12 PF KEYS
```

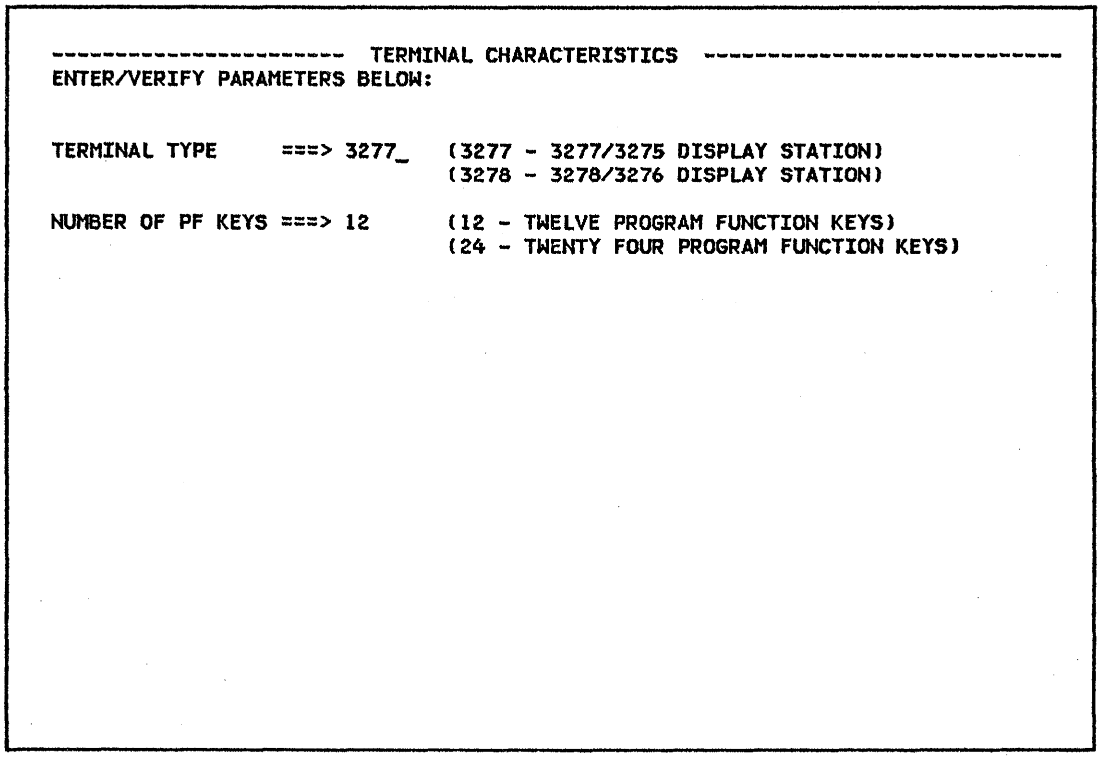
Figure 12. SPF parameter menu

The following describes each of the SPF parms functions, corresponding to the three options on the secondary menu.

SPECIFY TERMINAL CHARACTERISTICS (OPTION 0.1)

When this option is selected (SPF parms option 1), a menu is displayed (Figure 13) which allows the user to specify the terminal type and number of program function (PF) keys on the terminal.

After these parameters have been reviewed or changed, the user can press the End PF key to return to the previous menu.



```
----------------------- TERMINAL CHARACTERISTICS ---------------------------
ENTER/VERIFY PARAMETERS BELOW:


TERMINAL TYPE      ===> 3277_   (3277 - 3277/3275 DISPLAY STATION)
                                (3278 - 3278/3276 DISPLAY STATION)

NUMBER OF PF KEYS ===> 12       (12 - TWELVE PROGRAM FUNCTION KEYS)
                                (24 - TWENTY FOUR PROGRAM FUNCTION KEYS)
```

Figure 13. Terminal characteristics menu

SPF will automatically determine the terminal type during SPF initialization, and set it to the appropriate value. SPF will automatically set (or change) the number of PF keys in the following cases:

- If the terminal type is 3277, SPF will initialize the number of keys to 12.
- If the terminal type is 3278, SPF will initialize the number of keys to whatever was "remembered" from the user's last SPF session (for a new user, the number of keys is initialized to 12).
- If the user presses a PF key higher than 12, SPF will set the number of keys to 24.

SPF cannot sense the number of PF keys if the user switches from a 3278 with 24 PF keys to a 3278 with 12 PF keys. In this case, the user must inform SPF of the number of PF keys via option 0.1. Otherwise, the incorrect set of "remembered" key definitions will be used (see option 0.3).

SPECIFY CONSOLE, LOG, AND LIST DEFAULTS (OPTION 0.2)

When this option is selected (SPF parms option 2), a menu is displayed which allows the user to specify default processing for the virtual console and for the SPF log and list files, to be used when SPF is terminated via primary option "X". (Figure 14). If default processing options have not been specified, primary option "X" will cause the SPF termination menu to be displayed.

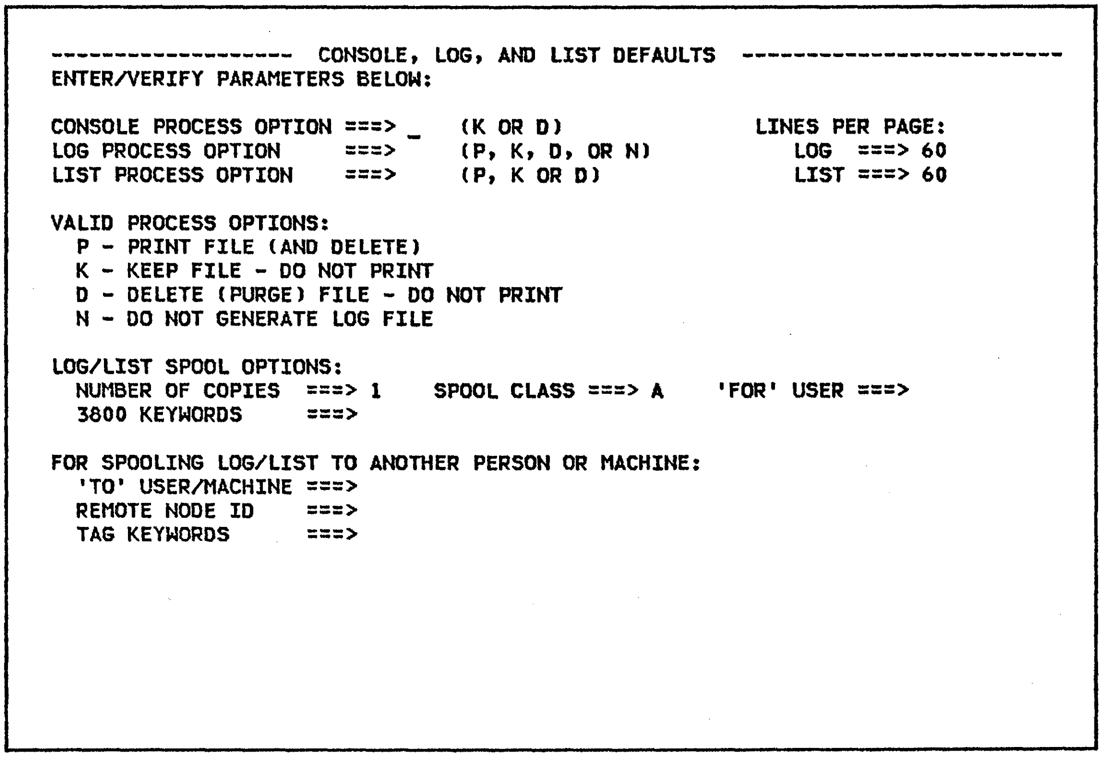
The number of lines per page and spool parameters for printing the log and list files may also be specified. The initial defaults for new users are:

```
Lines per page    ===> 60  (for both log and list)
Number of copies ===> 1
Spool class       ===> A
```

No other SPF-supplied initial defaults are provided for the other parameters on this menu.

Normal values for lines per page are:

```
60 - for printing 6 lines per inch
80 - for printing 8 lines per inch
```



```
------------------- CONSOLE, LOG, AND LIST DEFAULTS -------------------------
ENTER/VERIFY PARAMETERS BELOW:

CONSOLE PROCESS OPTION ===> _   (K OR D)                LINES PER PAGE:
LOG PROCESS OPTION     ===>     (P, K, D, OR N)            LOG  ===> 60
LIST PROCESS OPTION    ===>     (P, K OR D)                LIST ===> 60

VALID PROCESS OPTIONS:
  P - PRINT FILE (AND DELETE)
  K - KEEP FILE - DO NOT PRINT
  D - DELETE (PURGE) FILE - DO NOT PRINT
  N - DO NOT GENERATE LOG FILE

LOG/LIST SPOOL OPTIONS:
  NUMBER OF COPIES  ===> 1    SPOOL CLASS ===> A    'FOR' USER ===>
  3800 KEYWORDS     ===>

FOR SPOOLING LOG/LIST TO ANOTHER PERSON OR MACHINE:
  'TO' USER/MACHINE ===>
  REMOTE NODE ID    ===>
  TAG KEYWORDS      ===>
```

Figure 14. Console, log, and list defaults menu

The virtual console is automatically started the first time SPF invokes a CMS command (typically, a LINK or ACCESS executed automatically on behalf of the user). Upon termination of SPF, the console may be kept (in "start" status) or deleted (purged and set to "stop" status). An appropriate processing option for the console would be:

K - if the user normally runs with a virtual console
D - if the user normally runs without a virtual console.

The SPF log file is created the first time the user performs some action which results in a log message, such as saving edited data or submitting a job to the batch machine. The SPF list file is created the first time the user requests a print function.

For the log file, a process option of N may be specified to prevent generation of the log. Generation of the list file may be avoided by simply not requesting any print functions.

See section entitled "Termination Menu" on page 20 for a discussion of spool parameters which may be specified for the log and list files.

After the parameters on this menu have been reviewed or changed, the user can press the End PF key to return to the previous menu.

SPECIFY SPF PROGRAM FUNCTION KEYS (OPTION 0.3)

When this option is selected (SPF parms option 3), a menu is displayed which allows the user to specify the mapping of the PF keys into specific functions (Figure 15). Before selecting this option, the user should ensure that the correct terminal type (3277 or 3278) and number of PF keys (12 or 24) has been specified via option 0.1.

SPF maintains three different sets of key definitions for each user, as follows:

```
3277 (or 3275) with 12 PF keys
3278 (or 3276) with 12 PF keys
3278 (or 3276) with 24 PF keys
```

The appropriate set of key definitions will be invoked based on the terminal type and number of keys (as specified via option 0.1). The menu shown in the figure is for a 3278 with 24 PF keys.

The allowable SPF-defined key functions are listed at the bottom of the menu. See section entitled "Program Access and Function Keys" for a description of these functions. More than one key can be assigned to the same function. A function need not be assigned to a key, with one exception: There must be at least one End PF key assigned within the first 12 keys.

Entering a blank for any PF key definition causes the specified PF key to be assigned its SPF default. The default key definitions are discussed under "Program Access and Function Keys" and are also shown in Figure 15.

The user may equate a PF key to an edit or browse primary command, or to an edit line command, in the following manner:

>string Causes the specified PF key, when used in edit or browse, to simulate the entering of a primary command. Example:

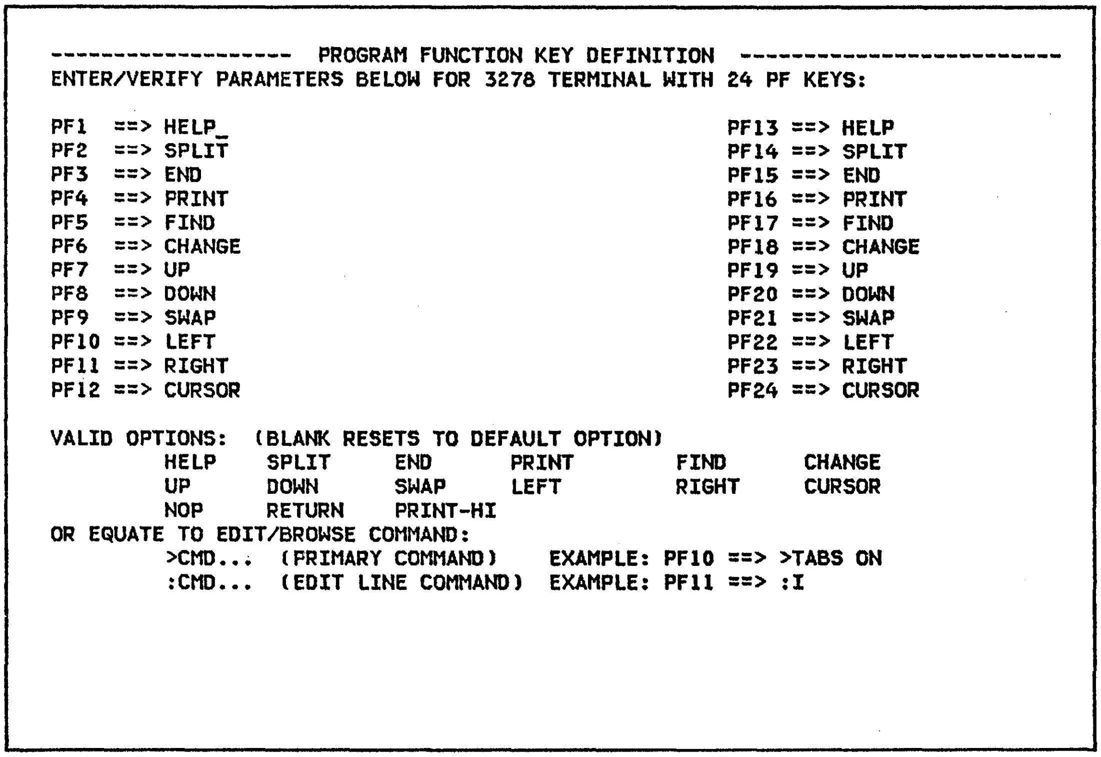
```
PF10 ===> >FIND ABC
```

Pressing PF10 when browsing or editing data will have the same effect as entering a FIND ABC primary command.

:string Causes the specified PF key, when used in edit, to simulate the entering of a line command. Example:

```
PF11 ===> :TF
```

Pressing PF11 when editing data will have the same effect as entering the "TF" (text flow) line command on whichever line the cursor is positioned when the key is pressed.



```
------------------- PROGRAM FUNCTION KEY DEFINITION -------------------------
ENTER/VERIFY PARAMETERS BELOW FOR 3278 TERMINAL WITH 24 PF KEYS:

PF1  ==> HELP_                                         PF13 ==> HELP
PF2  ==> SPLIT                                         PF14 ==> SPLIT
PF3  ==> END                                           PF15 ==> END
PF4  ==> PRINT                                         PF16 ==> PRINT
PF5  ==> FIND                                          PF17 ==> FIND
PF6  ==> CHANGE                                        PF18 ==> CHANGE
PF7  ==> UP                                            PF19 ==> UP
PF8  ==> DOWN                                          PF20 ==> DOWN
PF9  ==> SWAP                                          PF21 ==> SWAP
PF10 ==> LEFT                                          PF22 ==> LEFT
PF11 ==> RIGHT                                         PF23 ==> RIGHT
PF12 ==> CURSOR                                        PF24 ==> CURSOR

VALID OPTIONS:  (BLANK RESETS TO DEFAULT OPTION)
         HELP    SPLIT     END      PRINT        FIND      CHANGE
         UP      DOWN      SWAP     LEFT         RIGHT     CURSOR
         NOP     RETURN    PRINT-HI
OR EQUATE TO EDIT/BROWSE COMMAND:
         >CMD...  (PRIMARY COMMAND)    EXAMPLE: PF10 ==> >TABS ON
         :CMD...  (EDIT LINE COMMAND)  EXAMPLE: PF11 ==> :I
```

Figure 15. Program function key menu

BROWSE (OPTION 1)

The browse option allows the user to display source data and listings stored in SPF libraries or CMS files with the following charactersitics:

- Fixed or variable record formats
- Logical record lengths up to and including 32767 bytes

When browse is selected (primary option 1), a menu is displayed to allow the user to specify either an SPF library or CMS file id. A user id, virtual device address, LINK access mode, and read password may also be specified, if needed, for linking to another user's virtual disk. See Figure 16.

For SPF libraries, MACLIBs, and TXTLIBs, the user can supply the name of the member to be browsed, or can leave the member name blank to request a member list from which a member may be selected.

When the user has properly specified the data to be viewed, the first page of data is displayed. Two lines are reserved at the top of the screen for title information, short messages, command entry, and the scroll amount. The remainder of the screen contains the data (Figure 17).

Note: Under browse, any invalid (non-displayable) characters in the data are displayed as periods.

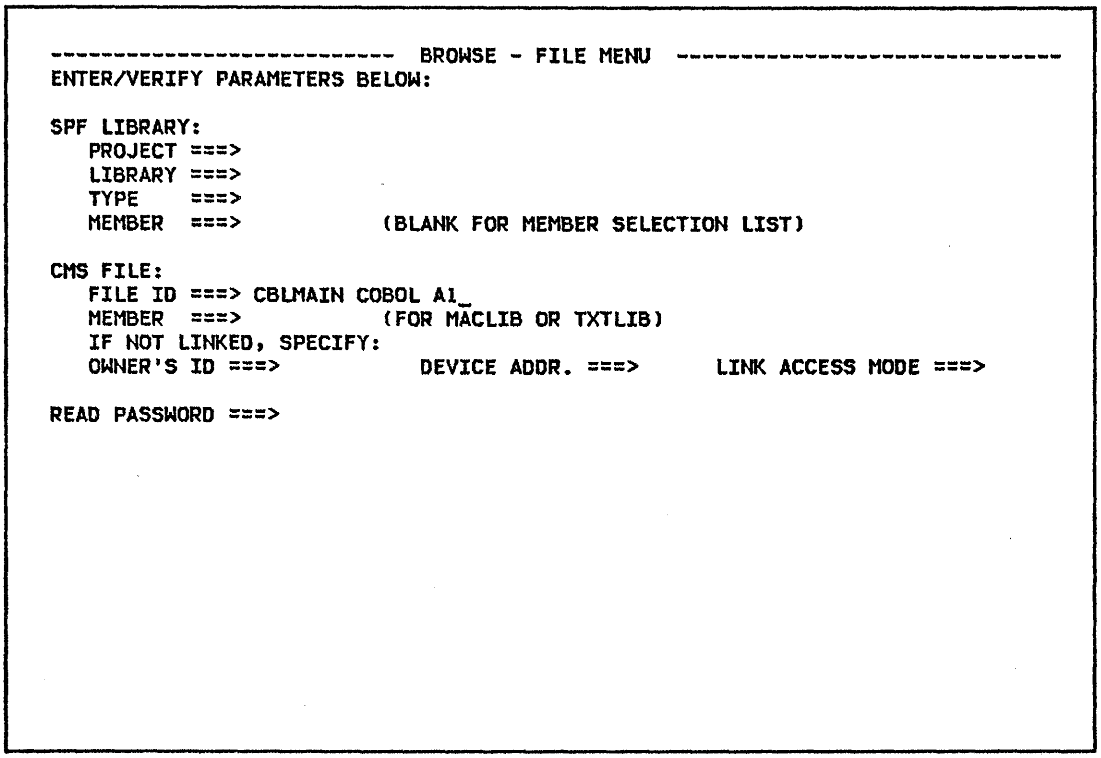
During browse, four-way scrolling is available via the Scroll PF keys. The FIND and LOCATE commands may also be used to scroll to a particular character string, line number, or symbolic label.

Browse provides six commands, described in the following sections, which may be entered in the command input field on line 2:

```
LOCATE          FIND
COLS            CAPS
RESET           HEX
```

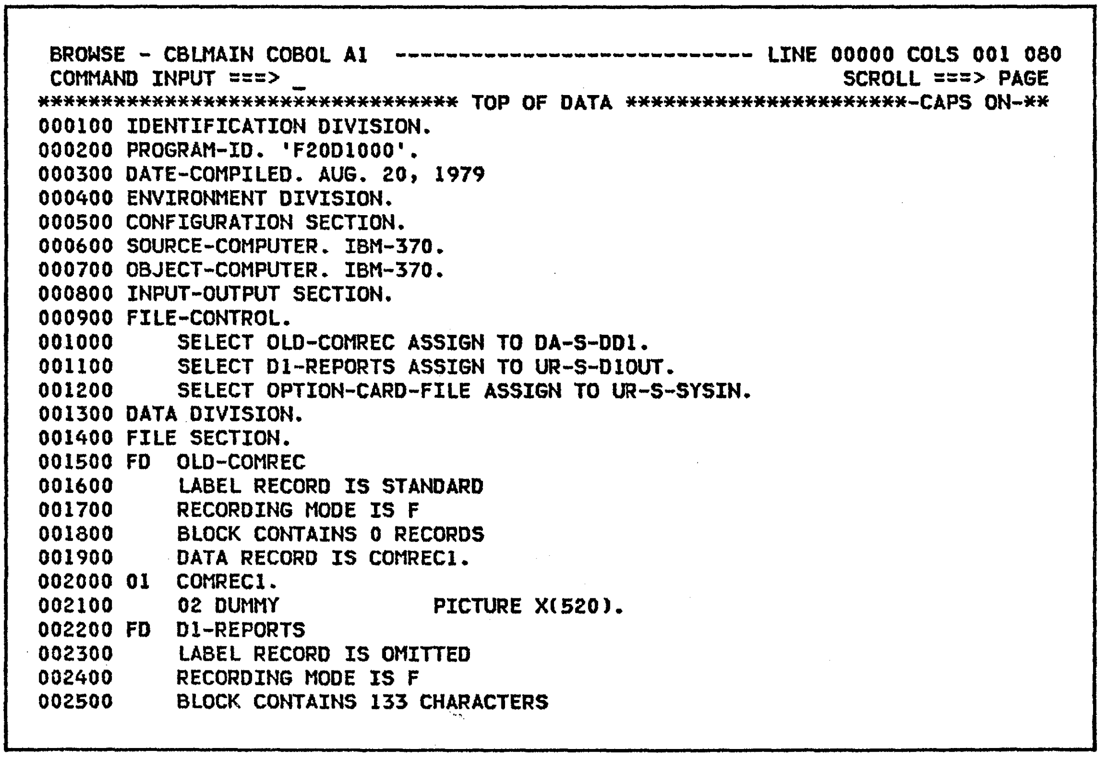
Browse may be terminated by pressing the End PF key, which causes a return to the previous menu (either the member selection list or the file menu). When return is to the member selection list, the member just browsed will appear at the top of the list. Another member may be selected from the list or the End key may be pressed again to return to the file menu.

When the file menu is displayed, another file or member may be selected, or the End key may be pressed to return to the primary option menu.



```
--------------------------- BROWSE - FILE MENU -----------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===>
   LIBRARY ===>
   TYPE    ===>
   MEMBER  ===>           (BLANK FOR MEMBER SELECTION LIST)

CMS FILE:
   FILE ID ===> CBLMAIN COBOL A1_
   MEMBER  ===>           (FOR MACLIB OR TXTLIB)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>
```

Figure 16. Browse - file menu



```
 BROWSE - CBLMAIN COBOL A1  ---------------------------- LINE 00000 COLS 001 080
 COMMAND INPUT ===> _                                        SCROLL ===> PAGE
******************************** TOP OF DATA **********************-CAPS ON-**
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. 'F20D1000'.
000300 DATE-COMPILED. AUG. 20, 1979
000400 ENVIRONMENT DIVISION.
000500 CONFIGURATION SECTION.
000600 SOURCE-COMPUTER. IBM-370.
000700 OBJECT-COMPUTER. IBM-370.
000800 INPUT-OUTPUT SECTION.
000900 FILE-CONTROL.
001000     SELECT OLD-COMREC ASSIGN TO DA-S-DD1.
001100     SELECT D1-REPORTS ASSIGN TO UR-S-D1OUT.
001200     SELECT OPTION-CARD-FILE ASSIGN TO UR-S-SYSIN.
001300 DATA DIVISION.
001400 FILE SECTION.
001500 FD  OLD-COMREC
001600     LABEL RECORD IS STANDARD
001700     RECORDING MODE IS F
001800     BLOCK CONTAINS 0 RECORDS
001900     DATA RECORD IS COMREC1.
002000 01  COMREC1.
002100     02 DUMMY            PICTURE X(520).
002200 FD  D1-REPORTS
002300     LABEL RECORD IS OMITTED
002400     RECORDING MODE IS F
002500     BLOCK CONTAINS 133 CHARACTERS
```

Figure 17. Browse - data display

LOCATING LINES

During browse, the current position of the screen window is indicated by the line/column numbers in the upper right hand corner of the screen. The line number refers to the first data line currently being displayed (i.e., the first line after the two header lines). It indicates the relative position of that line in the file (or member), and is unrelated to any sequence numbers which may be contained within the data. The "TOP OF DATA" message is treated as relative line zero.

The LOCATE command may be used to scroll directly to a specified line number. It may also be used to scroll to a symbolic label, which must have been previously defined by the user, as follows:

Scroll to any desired line and enter a label of the form:

```
.xxxxxxx
```

(period followed by up to seven alphameric characters) in the command input line. The label is treated as an internal symbol and equated to the top line on the screen. Labels are not retained upon exit from browse.

Format of the LOCATE command:

```
LOCATE  line-number/label

Command abbreviations: LOC, L
```

Either a line number or a label must be entered as an operand.

The "line-number" operand is a numeric value of up to eight digits (leading zeros need not be typed), which refers to the relative line number as displayed in the upper right hand corner.

The "label" operand consists of a period (.) immediately followed by up to seven alphameric characters. Scrolling is performed to the line which was previously equated to the label, as described above.

Note: The period is mandatory when defining a label. It is optional when referring to a label in a LOCATE command unless the label begins with a numeric character, in which case the period is required to distinguish the label from a line number.

IDENTIFYING COLUMNS

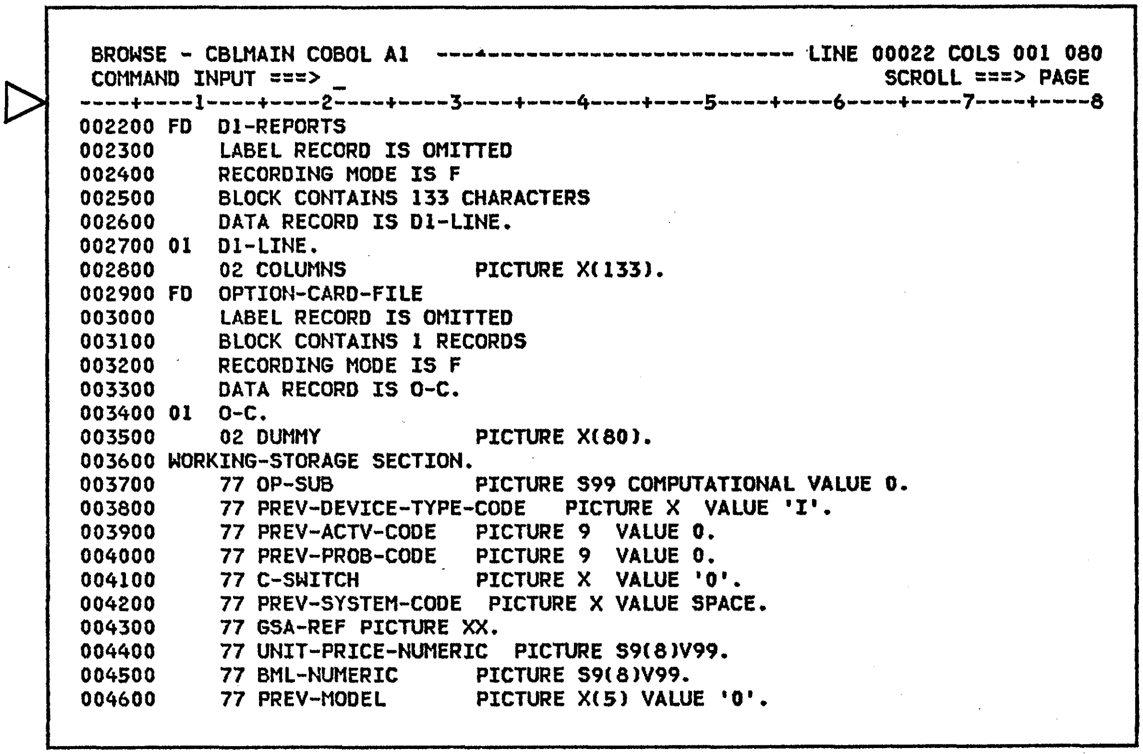
The COLS command causes a column identification line to be displayed, temporarily overlaying the first data line. The RESET command terminates display of the column identification line. Formats:

COLS
Command abbreviation: COL

RESET
Command abbreviation: RES

Neither command uses any parameters.

An example of the column identification line is shown in Figure 18. The digits on the identification line indicate the "tens" positions, e.g., "1" indicates column 10, "2" indicates column 20, etc. The plus signs (+) indicate the "fives" positions.



```
 BROWSE - CBLMAIN COBOL A1  ----------------------------- LINE 00022 COLS 001 080
 COMMAND INPUT ===> _                                            SCROLL ===> PAGE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
002200 FD  D1-REPORTS
002300      LABEL RECORD IS OMITTED
002400      RECORDING MODE IS F
002500      BLOCK CONTAINS 133 CHARACTERS
002600      DATA RECORD IS D1-LINE.
002700 01  D1-LINE.
002800      02 COLUMNS          PICTURE X(133).
002900 FD  OPTION-CARD-FILE
003000      LABEL RECORD IS OMITTED
003100      BLOCK CONTAINS 1 RECORDS
003200      RECORDING MODE IS F
003300      DATA RECORD IS O-C.
003400 01  O-C.
003500      02 DUMMY            PICTURE X(80).
003600 WORKING-STORAGE SECTION.
003700      77 OP-SUB           PICTURE S99 COMPUTATIONAL VALUE 0.
003800      77 PREV-DEVICE-TYPE-CODE   PICTURE X  VALUE 'I'.
003900      77 PREV-ACTV-CODE   PICTURE 9  VALUE 0.
004000      77 PREV-PROB-CODE   PICTURE 9  VALUE 0.
004100      77 C-SWITCH         PICTURE X  VALUE '0'.
004200      77 PREV-SYSTEM-CODE  PICTURE X VALUE SPACE.
004300      77 GSA-REF PICTURE XX.
004400      77 UNIT-PRICE-NUMERIC  PICTURE S9(8)V99.
004500      77 BML-NUMERIC      PICTURE S9(8)V99.
004600      77 PREV-MODEL       PICTURE X(5) VALUE '0'.
```

Figure 18. Browse - column identification line

FINDING CHARACTER STRINGS

The FIND command is used to find one or more occurrances of a specified character string. It allows special forms of character strings and several optional parameters to control:

- Starting point and direction of search
- Special conditions to control character string "match"
- Column limitations on extent of search

The command summary, which follows, describes the basic form of FIND which is most frequently needed. The remainder of this section provides detailed information about the various special features and options.

Command Summary

The basic format for FIND is:

FIND string [ALL]
Command abbreviation: F

The operands may be separated by blanks or commas. The ALL operand is optional, and may precede or follow the string.

As a general rule, the strings are typed without delimiters. For example:

```
FIND  XYZ
```

to find the next occurrence of XYZ.

Strings may be delimited with either apostrophes (') or quotation marks ("). Delimiters must be used if a string contains imbedded blanks or commas, or if a string is the same as a command keyword. For example:

```
FIND  'EVERY ONE'
```

to find the next occurrence of "EVERY ONE" (which contains a blank).

If the ALL operand is omitted, the FIND command searches for the next occurrence of the string starting at the current cursor location. (If the cursor is not in the data area of the screen, scanning starts at the beginning of the first line which is currently being displayed.) Scrolling is performed, if necessary, to bring the string into view. The cursor is positioned under the string and a verification message is displayed in the upper right-hand corner of the screen. The Find PF key may be used to find each successive occurrence of the string.

If the string is not found between the current cursor location and the end of data, a "BOTTOM OF DATA REACHED" message is displayed and the audible alarm is sounded (if installed). The Find PF key may then be used to wrap to the top of data and continue searching. If the string is not found anywhere in the data a "NO xxxxx FOUND" message is displayed.

If the ALL operand is typed, the FIND command searches for all occurrences of string starting at the top of the data, and positions the cursor under the first occurrence. The verification message indicates the number of occurrences found.

Detailed Command Description

The complete format for FIND, showing all optional parameters, follows:

```
FIND  string  [NEXT/ALL/FIRST/LAST/PREV]
              [CHARS/PREFIX/SUFFIX/WORD]
              [col-1 [col-2]]

Command abbreviation:  F
Keyword abbreviations: PRE (for PREFIX), SUF (for suffix)
```

The operands may be separated with blanks or commas, and may be typed in any order except that col-2, if typed, must follow col-1. The string operand is required; the others are optional.

The "string" operand specifies the string to be found. It may be specified in any one of the following forms:

- Any string of characters not starting or ending with an apostrophe or quotation mark, and not containing any imbedded blanks or commas.

- A delimited string: Any string starting and ending with an apostrophe (') but not containing imbedded apostrophes, or starting and ending with a quotation mark (") but not containing imbedded quotation marks.

- A hex string: Any delimited string of valid hexadecimal characters, preceded or followed with the character X. Example: X'C27B'

- A text string: Any delimited string of characters, preceded or followed with the character T. See discussion under "Use of Text Strings." Example: T'conditions for'

- A picture string: Any delimited string of picture characters, preceded or followed with the character P. See discussion under "Use of Picture Strings." Example: P'.'

- A single asterisk (*). This causes the previous value of "string" that was used in a FIND command to be used again. The previous value of a character string (referenced via * or used by the Find PF key) is retained until the browse option is terminated (i.e. until return to the primary option menu).

Starting Point and Direction of Search

The starting point, direction and extent of the search may be controlled by one of the following operands.

NEXT   Scan starts at the current cursor location and proceeds forwards to find the next occurrence of the string.

ALL   Scan starts at the top of data and proceeds forwards to find all occurrences of the string.

FIRST Scan starts at the top of data and proceeds forwards to find the first occurrence of the string.

LAST   Scan starts at the bottom of data and proceeds backwards to find the last occurrence of the string.

PREV   Scan starts at the current cursor location and proceeds backwards to find the previous occurrence of the string.

If this operand is omitted, the default is NEXT.

If the direction of the search is forward (i.e, if FIRST, ALL, or NEXT was specified), pressing the Find PF key finds the next occurrence of the designated string. If the direction of the search is backward (i.e., if LAST or PREV was specified), pressing the Find PF key finds the previous occurrence of the designated string. The other optional parameters (CHARS, WORD, PREFIX, SUFFIX, col-1, and col-2) remain in effect, as specified on the last FIND command.

The search proceeds until one or all occurrences of the string are found, or until the end of data is encountered. If the string is not found, one of the following actions takes place.

- For FIRST, LAST, or ALL - a "NO xxxxx FOUND" message is displayed in the upper right-hand corner of the screen.
- For NEXT - a "BOTTOM OF DATA REACHED" message is displayed.
- For PREV - a "TOP OF DATA REACHED" message is displayed.

When "BOTTOM OF DATA REACHED" or "TOP OF DATA REACHED" is displayed, the user may press the Find PF key to continue searching by wrapping to the top (or bottom) of the data. If no occurrence is found anywhere in the data, a "NO xxxxx FOUND" message is displayed.

Conditions for Character String Match

The conditions for a successful "match" with the string may be controlled based on whether the data string begins and/or ends with a non-alphameric character (i.e., a special character or blank). The operands are: CHARS, PREFIX (may be abbreviated PRE), SUFFIX (may be abbreviated SUF), and WORD. In the following illustration, the underscored strings would be found, and the non-underscored strings would be ignored (skipped over).

CHARS 'DO'	-	D̲O̲	D̲O̲NT	AD̲O̲	AD̲O̲PT	'D̲O̲'	+AD̲O̲	(D̲O̲NT)	AD̲O̲-
PREFIX 'DO'	-	DO	D̲O̲NT	ADO	ADOPT	'DO'	+ADO	(D̲O̲NT)	ADO-
SUFFIX 'DO'	-	DO	DONT	AD̲O̲	ADOPT	'DO'	+AD̲O̲	(DONT)	AD̲O̲-
WORD 'DO'	-	D̲O̲	DONT	ADO	ADOPT	'D̲O̲'	+ADO	(DONT)	ADO-

If this operand is omitted, the default is CHARS.

Column Limitations

If the "col-1" and "col-2" operands are typed (two integers separated by a comma or at least one blank), they indicate the starting and ending column positions for the search. If col-1 is specified without col-2, the string will be found only if it starts in the specified column. If neither col-1 nor col-2 is specified, the search will continue across all columns.

Use of Text Strings

A text string, which may be used as the string in a FIND command, allows the search to be satisfied without regard to upper/lower case alphabetics. Example: FIND T'CONDITION NO. 1' would successfully find any of the following:

```
CONDITION NO. 1
Condition No. 1
condition no. 1
coNDitION nO. 1
```

An ordinary delimited string, such as: FIND 'CONDITION NO. 1' would find only the first example listed above.

Within the text string itself, is is immaterial whether alphabetics are typed in upper or lower case (even if CAPS mode is off). For example, all the following have the same effect:

```
FIND T'Edit Commands'
FIND T'EDIT COMMANDS'
FIND T'edit commands'
```

Use of Picture Strings

A picture string, which may be used as the string in a FIND command, allows the search to be satisfied when a particular type of character is encountered. Special characters are used within the picture string to represent the type of character to be found, as follows:

STRING		MEANING
P'='	-	any character (don't care)
P'¬'	-	any non-blank character
P'.'	-	any non-displayable (invalid) character
P'#'	-	any numeric character (0-9)
P'-'	-	any non-numeric character
P'@'	-	any alphabetic character (upper or lower case)
P'<'	-	any lower case alphabetic character
P'>'	-	any upper case alphabetic character
P'$'	-	any special character (not alpha or numeric)

Any special characters other than the ones listed above are invalid within a picture string. A picture string may also include any alphabetic or numeric characters, in which case the characters represent themselves.

Examples of picture strings:

```
P'###'   - a string of three numeric characters
P'¬ ¬'   - any two non-blank characters surrounding a blank
P'.'     - any non-displayable character
P' #'    - a blank followed by a numeric character
P'#AB'   - a numeric character followed by 'AB'
```

Examples of FIND command using picture strings:

```
FIND P'.'      - find the next non-displayable character
FIND P'¬' 72   - find the next non-blank character in col 72
F P' ¬'  1     - find the next line with a blank in col 1
                 followed by a non-blank
```

When the picture string P'.' is used to find a non-displayable character, the hexadecimal representation of the character is shown in the confirmation message which appears in the upper right-hand corner of the screen.

CHARACTER STRING TRANSLATION

If the character string operand of the FIND command contains alphabetic characters, they are automatically translated to upper case if browse is operating with CAPS mode on; they are left as-is (not translated) if browse is operating with CAPS mode off.

CAPS mode may be turned on or off by using the CAPS command. Format:

```
   CAPS  [ON/OFF]
```

If the ON/OFF operand is omitted, ON is the default. Under browse, CAPS mode is always initialized ON.

Note: The "TOP OF DATA" and "BOTTOM OF DATA" lines contain an indication of the current mode.

HEXADECIMAL DISPLAY

When browse is operating in HEX mode, three lines are displayed for each source line. The first line shows the data in standard character form. The following two lines show the same data in hexadecimal representation. See Figure 19.

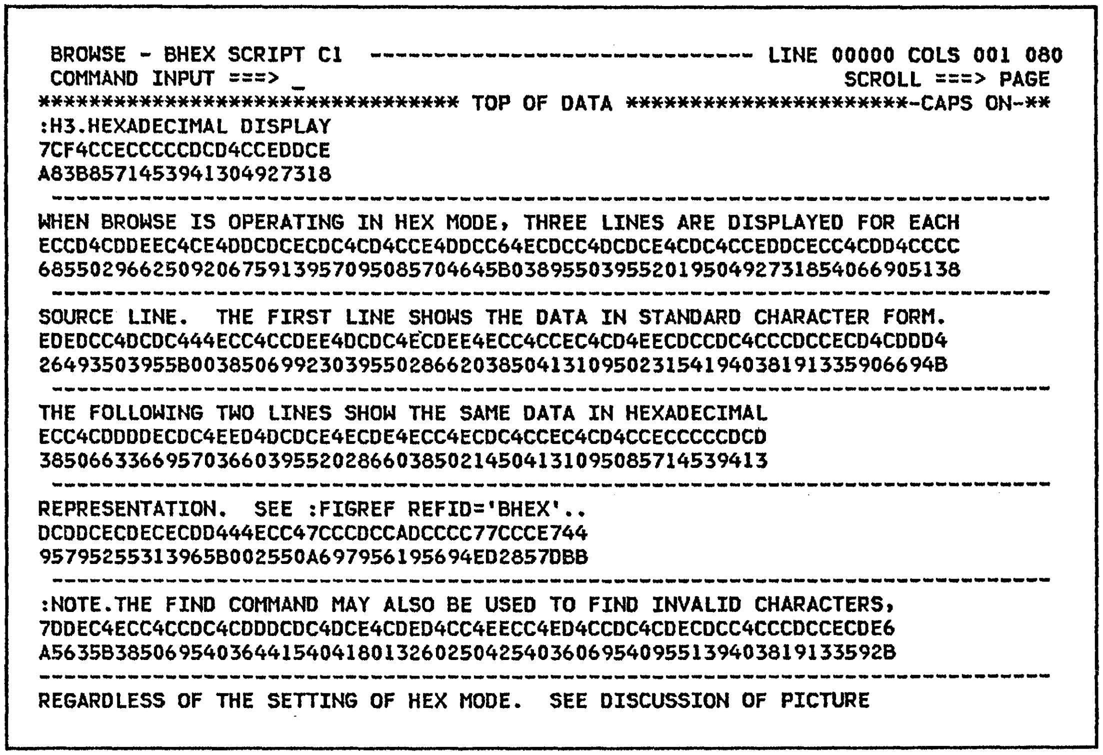
Note: The FIND command may also be used to find invalid characters or any specific hex character, regardless of the setting of HEX mode. See discussion of picture strings and hex strings under the FIND command.

HEX mode may be turned on or off by using the HEX command. Format:

```
   HEX  [ON/OFF]  [VERT/DATA]
```

The operands may be typed in any order. If the ON/OFF operand is omitted, ON is the default.

The VERT (vertical) and DATA operands are valid only when HEX mode is turned on. VERT causes the hexadecimal representations to be displayed vertically (two rows per byte) under each character, as shown in Figure 19. DATA causes the hexadecimal representations to be displayed as a string of hex characters (two per byte). Since the hex string is twice as long as the data string, it consumes two rows. If this operand is omitted, VERT is the default.



```
 BROWSE - BHEX SCRIPT C1  ----------------------------- LINE 00000 COLS 001 080
 COMMAND INPUT ===> _                                          SCROLL ===> PAGE
********************************* TOP OF DATA ***********************-CAPS ON-**
:H3.HEXADECIMAL DISPLAY
7CF4CCECCCCCDCD4CCEDDCE
A83B8571453941304927318
 ------------------------------------------------------------------------------
WHEN BROWSE IS OPERATING IN HEX MODE, THREE LINES ARE DISPLAYED FOR EACH
ECCD4CDDEEC4CE4DDCDCECDC4CD4CCE4DDCC64ECDCC4DCDCE4CDC4CCEDDCECC4CDD4CCCC
685502966250920675913957095085704645B03895503955201950492731854066905138
 ------------------------------------------------------------------------------
SOURCE LINE.  THE FIRST LINE SHOWS THE DATA IN STANDARD CHARACTER FORM.
EDEDCC4DCDC444ECC4CCDEE4DCDC4ECDEE4ECC4CCEC4CD4EECDCCDC4CCCDCCECD4CDDD4
26493503955B0038506992303955028662038504131095023154194038191335906694B
 ------------------------------------------------------------------------------
THE FOLLOWING TWO LINES SHOW THE SAME DATA IN HEXADECIMAL
ECC4CDDDDECDC4EED4DCDCE4ECDE4ECC4ECDC4CCEC4CD4CCECCCCCDCD
385066336695703660395520286603850214504131095085714539413
 ------------------------------------------------------------------------------
REPRESENTATION.  SEE :FIGREF REFID='BHEX'..
DCDDCECDECECDD444ECC47CCCDCCADCCCC77CCCE744
957952553139658002550A697956195694ED2857DBB
 ------------------------------------------------------------------------------
:NOTE.THE FIND COMMAND MAY ALSO BE USED TO FIND INVALID CHARACTERS,
7DDEC4ECC4CCDC4CDDDCDC4DCE4CDED4CC4EECC4ED4CCDC4CDECDCC4CCCDCCECDE6
A5635B385069540364415404180132602504254036069540955139403819133592B
 ------------------------------------------------------------------------------
REGARDLESS OF THE SETTING OF HEX MODE.  SEE DISCUSSION OF PICTURE
```

Figure 19. Browse - hexadecimal display, vertical

EDIT (OPTION 2)

The edit option allows the user to create, display, and modify source data (program code, test data, documentation, etc.) stored in SPF libraries or CMS files with the following characteristics:

- Fixed or variable record formats
- Logical record lengths up to and including 255 bytes, but not less than 10 bytes

When edit is selected (primary option 2), a menu is displayed to allow the user to specify either an SPF library or CMS file id. A user id, virtual device address, and link access mode may also be specified, if needed, for linking to another user's virtual disk. See Figure 20.

For SPF libraries and CMS files that are MACLIBs or TXTLIBs, the user can supply the name of the member to be edited, or leave the member name blank to request a member list from which a member may be selected.

If a CMS file id is specified and the file does not currently exist, the record format (RECFM) and logical record length (LRECL) may be specified on the menu. If these fields are left blank, the RECFM and LRECL which the user last specified for this filetype will be used. If the user has never specified a RECFM and LRECL for this filetype, the new file will be created with RECFM=F (fixed) and LRECL=80.

If the minidisk is password protected, an update password must be specified. For a concatenated sequence of SPF libraries which do not reside on the same minidisk, two passwords may be required: An update password for the first library in the concatenation sequence, and a read password for the other libraries (which must all be accessable via the same password).

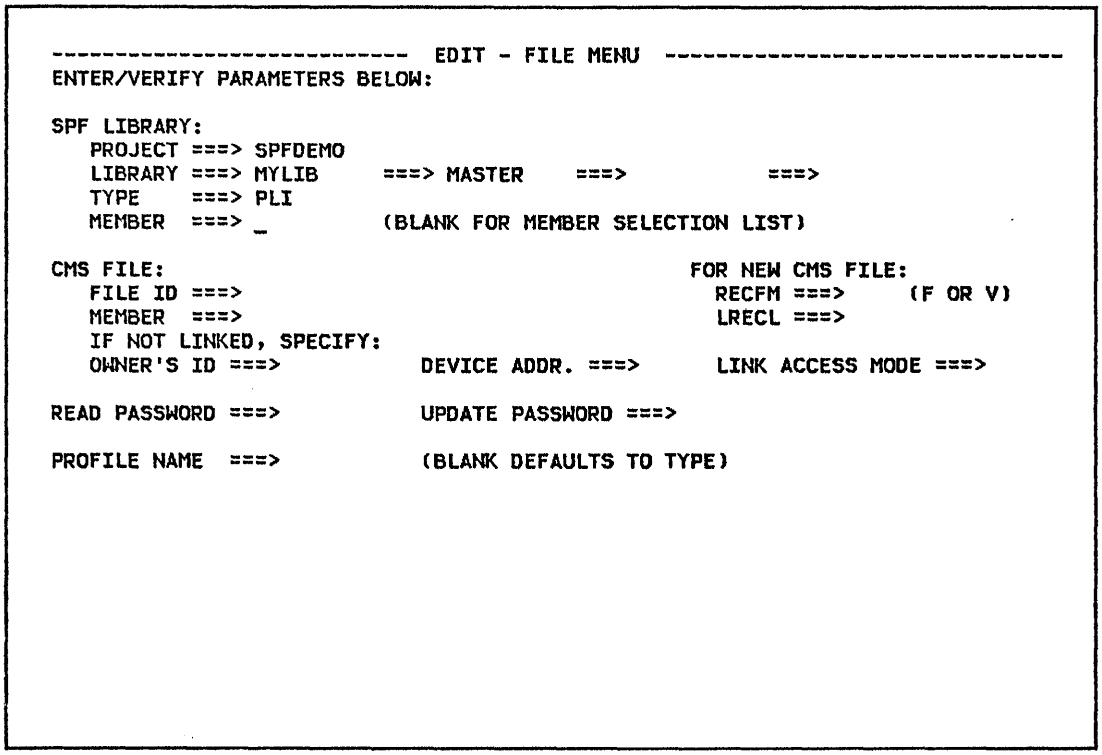
The menu also allows specification of a profile name, which may be entered to override the default edit profile. See description under "Edit Modes and Profiles."

A concatenated sequence of SPF libraries may be specified. The concatenated sequence applies to fetching of members to be edited. The libraries are searched in the designated order to find the member and bring it into working storage. If a member selection list is requested, the list indicates the library where the member was found (library 1, 2, 3, or 4) -- see Figure 21. When the edited member is saved, it is placed (or replaced) in the first library in the concatenation sequence regardless of its original location.

SPF libraries must have been previously specified to SPF (see "File Utility", option 3.2). Selection of a non-existent CMS sequential file or a non-existent member of a library allows creation of new data.

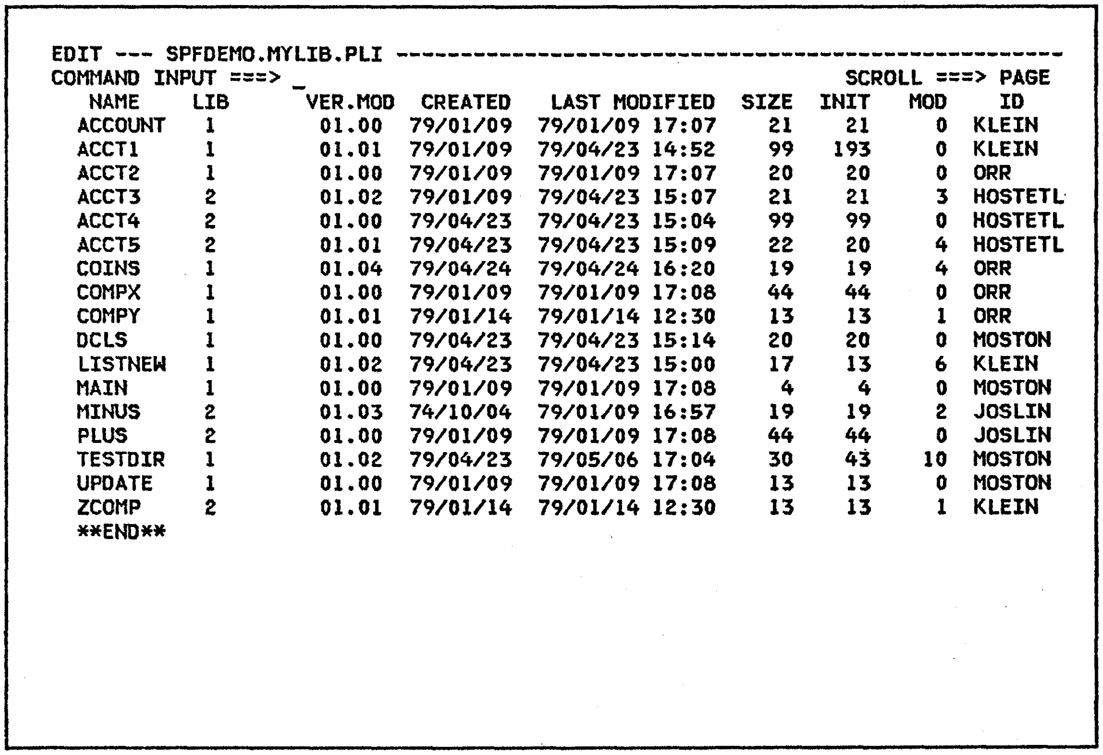
The member or sequential file is read into virtual storage, wherein it is updated during the edit operations. Use of virtual storage for edit work space results in high performance, but requires sufficient virtual storage for the data being edited. Warning messages will be displayed if the amount of available virtual storage drops below 50K bytes. If all available storage is consumed an ABEND will occur, and the user must then re-IPL CMS.

The edit data display is similar to a browse display except that each line consists of a six-column line-number field followed by a 72-column data field (see Figure 22). The line-number field reflects the contents of the sequence numbers in the data, if the data is numbered. For unnumbered data, the line numbers start at 1 and are incremented by 1.



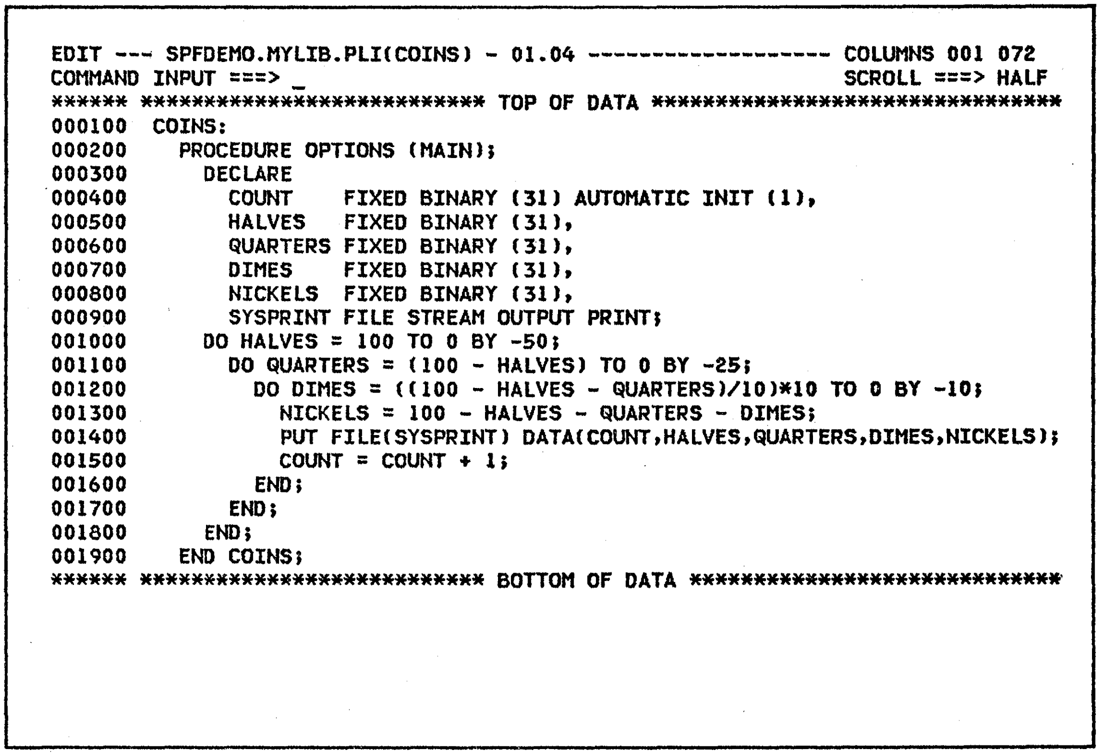
```
---------------------------- EDIT - FILE MENU ------------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB     ===> MASTER    ===>           ===>
   TYPE    ===> PLI
   MEMBER  ===> _         (BLANK FOR MEMBER SELECTION LIST)

CMS FILE:                                         FOR NEW CMS FILE:
   FILE ID ===>                                     RECFM ===>     (F OR V)
   MEMBER  ===>                                     LRECL ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>           UPDATE PASSWORD ===>

PROFILE NAME  ===>           (BLANK DEFAULTS TO TYPE)
```

Figure 20. Edit - file menu



```
EDIT --- SPFDEMO.MYLIB.PLI -------------------------------------------------------
COMMAND INPUT ===> _                                          SCROLL ===> PAGE
   NAME     LIB      VER.MOD  CREATED    LAST MODIFIED    SIZE  INIT   MOD    ID
  ACCOUNT    1        01.00  79/01/09  79/01/09 17:07     21    21     0  KLEIN
  ACCT1      1        01.01  79/01/09  79/04/23 14:52     99   193     0  KLEIN
  ACCT2      1        01.00  79/01/09  79/01/09 17:07     20    20     0  ORR
  ACCT3      2        01.02  79/01/09  79/04/23 15:07     21    21     3  HOSTETL
  ACCT4      2        01.00  79/04/23  79/04/23 15:04     99    99     0  HOSTETL
  ACCT5      2        01.01  79/04/23  79/04/23 15:09     22    20     4  HOSTETL
  COINS      1        01.04  79/04/24  79/04/24 16:20     19    19     4  ORR
  COMPX      1        01.00  79/01/09  79/01/09 17:08     44    44     0  ORR
  COMPY      1        01.01  79/01/14  79/01/14 12:30     13    13     1  ORR
  DCLS       1        01.00  79/04/23  79/04/23 15:14     20    20     0  MOSTON
  LISTNEW    1        01.02  79/04/23  79/04/23 15:00     17    13     6  KLEIN
  MAIN       1        01.00  79/01/09  79/01/09 17:08      4     4     0  MOSTON
  MINUS      2        01.03  74/10/04  79/01/09 16:57     19    19     2  JOSLIN
  PLUS       2        01.00  79/01/09  79/01/09 17:08     44    44     0  JOSLIN
  TESTDIR    1        01.02  79/04/23  79/05/06 17:04     30    43    10  MOSTON
  UPDATE     1        01.00  79/01/09  79/01/09 17:08     13    13     0  MOSTON
  ZCOMP      2        01.01  79/01/14  79/01/14 12:30     13    13     1  KLEIN
  **END**
```

Figure 21. Edit - member selection list



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------ COLUMNS 001 072
COMMAND INPUT ===> _                                         SCROLL ===> HALF
****** ************************** TOP OF DATA *******************************
000100  COINS:
000200    PROCEDURE OPTIONS (MAIN);
000300      DECLARE
000400        COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500        HALVES   FIXED BINARY (31),
000600        QUARTERS FIXED BINARY (31),
000700        DIMES    FIXED BINARY (31),
000800        NICKELS  FIXED BINARY (31),
000900        SYSPRINT FILE STREAM OUTPUT PRINT;
001000      DO HALVES = 100 TO 0 BY -50;
001100        DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200          DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300            NICKELS = 100 - HALVES - QUARTERS - DIMES;
001400            PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001500            COUNT = COUNT + 1;
001600          END;
001700        END;
001800      END;
001900    END COINS;
****** ************************** BOTTOM OF DATA ****************************
```

Figure 22. Edit - data display

Notes:

1. Under edit, any invalid (non-displayable) characters are replaced on the screen (but not in the data) with attribute bytes, which display as blanks and may not be overtyped.

2. Invalid characters may be displayed and edited by entering HEX mode (see description of HEX command) or by using the FIND and CHANGE commands.

SUMMARY OF EDIT OPERATIONS

Under edit, four-way scrolling is available via the Scroll PF keys. The FIND and LOCATE commands may also be used to scroll to a particular character string or line number.

To modify one or more lines of data, the user simply moves the cursor to the desired location and enters the new information by overtyping the existing lines. Several lines may be modified before pressing the ENTER key.

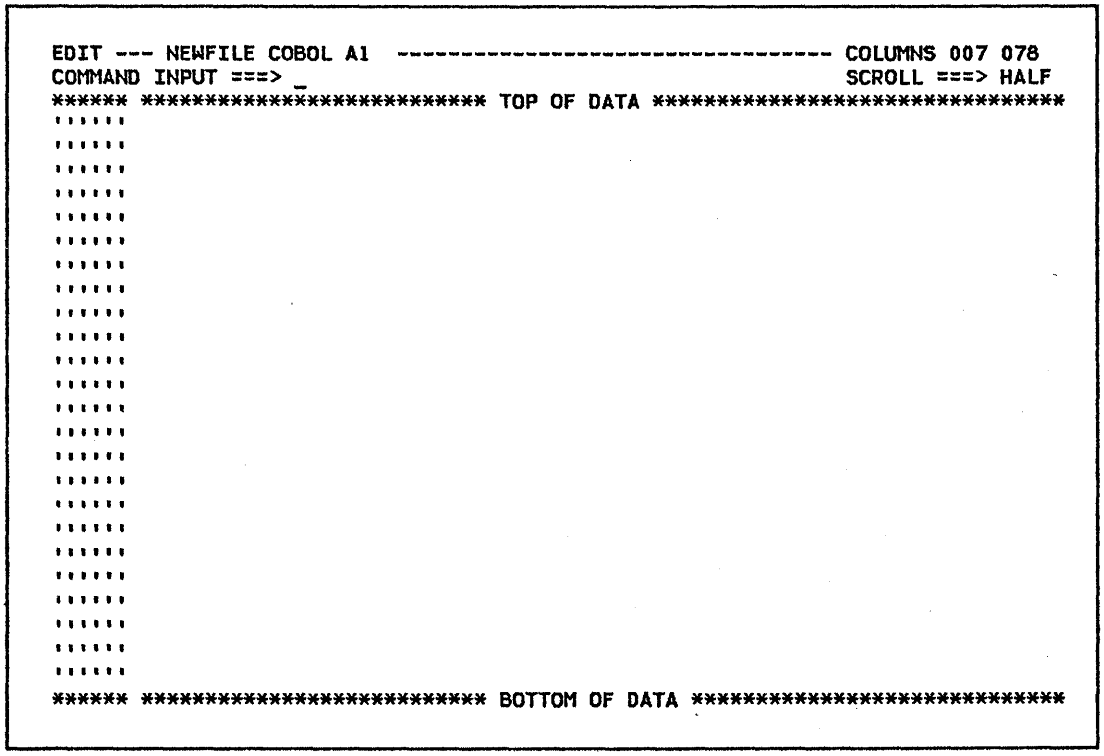
Edit Commands

Lines may be deleted, inserted, shifted left or right (for indentation changes), duplicated, or rearranged by overtyping the line-number fields with "line commands" consisting of one or more characters. Single character line commands operate on individual lines (e.g., D to delete a line, I to insert a blank line, M to move a line). Double character line commands operate on blocks of lines (e.g., DD on two different lines to indicate the first and last lines to be deleted).

In most cases, a number may follow the line command to indicate multiple occurrences (e.g., I3 to insert three blank lines). Several line commands as well as data modifications may be typed before pressing the ENTER key.

__Note__: SPF does not distinguish between "input mode" and "edit mode" during edit operations. The I (insert) line command is used to add new lines, either between existing lines or at the end of the data. When a new file or member is specified for editing, the initial display contains several inserted lines between the "TOP OF DATA" and "BOTTOM OF DATA" message lines (Figure 23).

For general edit operations, "primary commands" may be entered at the top of the screen to: control edit modes, locate a specific line, submit data to the job stream of another virtual machine, find and change designated character strings, control sequence numbering and character translation, effect segmentation changes (splitting a member into two members or merging two members into one), save the edited data, or cancel without saving.



```
EDIT --- NEWFILE COBOL A1  ---------------------------------- COLUMNS 007 078
COMMAND INPUT ===> _                                          SCROLL ===> HALF
****** *************************** TOP OF DATA *********************************
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** *************************** BOTTOM OF DATA ******************************
```

Figure 23. Edit - creating new data

A new SPF user need not master all the line commands and primary commands to make effective use of edit. The command descriptions in this manual are grouped under the following headings:

- __Basic Line Commands__ - Describes the commands used for identifying columns, inserting and deleting lines, repeating lines, moving and copying lines, and rules for entering line commands. Should be read by all users.

- __General Commands__ - Describes the commands used for locating lines, sequence number generation and control, statistics generation, automatic source listings, recovery control, general reset, submit to job stream, save and cancel. Should be read by all users.

- Finding and Changing Character Strings - Contains a summary of the FIND and CHANGE commands, which should be read by all users, followed by a detailed description of FIND/CHANGE optional parameters and features.

- Advanced Features - Describes the commands used for line mask definition, indentation changes, boundary column definition, excluded lines, segmentation changes, nulls control, tabs definition and control, profile display and control, and hexadecimal display.

- Text Preparation Features - Describes the commands used for character translation, text entry and edit, and overlaying lines.

A list of edit commands, and the section under which each is described, is shown in Figure 24. A command summary may also be found in Appendix D, and a quick reference summary in Appendix E.

DOCUMENT SECTION	EDIT LINE COMMANDS		EDIT PRIMARY COMMANDS	
BASIC LINE COMMANDS	COLS I - INSERT D - DELETE R - REPEAT	M - MOVE C - COPY A - AFTER B - BEFORE		
GENERAL COMMANDS			LOCATE NUMBER RENUM UNNUM AUTONUM STATS	PRINT RECOVERY RESET SUBMIT SAVE CANCEL
FIND AND CHANGE			FIND CHANGE	
ADVANCED FEATURES	MASK < - DATA LEFT > - DATA RIGHT (- COLS LEFT) - COLS RIGHT BOUNDS	X - EXCLUDE S - SHOW F - FIRST L - LAST TABS	COPY MOVE CREATE REPLACE	NULLS TABS PROFILE HEX
TEXT PREPARATION FEATURES	TE - TEXT ENTRY TS - TEXT SPLIT TF - TEXT FLOW O - OVERLAY		CAPS	

Figure 24. List of edit commands

Edit Modes and Profiles

Edit is capable of operating in the following modes:

- NUMBER Mode - controls validation and generation of sequence numbers.
- AUTONUM Mode - controls automatic renumbering when the data is saved.
- STATS Mode - controls generation of SPF statistics for members of SPF libraries.
- PRINT Mode - controls automatic recording of source listings in the SPF list file.
- CAPS Mode - controls translation of alphabetic characters to upper case.
- NULLS Mode - controls replacement of trailing blanks with null characters on the display screen.
- TABS Mode - controls use of "hardware" or "logical" tabs feature.
- HEX Mode - controls display of data in hexadecimal representation.
- RECOVERY Mode - controls automatic recording of edit transactions to facilitate recovery following a system failure.

Each mode may be turned on or off independently of the other modes by means of the following primary commands: NUMBER, AUTONUM, STATS, PRINT, CAPS, NULLS, TABS, HEX, and RECOVERY. See the command descriptions for further explanation.

The current settings of the modes, together with the current MASK, TABS, and BOUNDS lines, are maintained in an edit "profile" which may be displayed at any time via the PROFILE primary command. When edit is terminated, the current profile is automatically saved in the user's SPFPARMS file.

SPF maintains up to 25 different edit profiles for each user. This allows different mode settings and different MASK, TABS, and BOUNDS lines to be remembered and used as the initial settings for different types of source data.

Each profile is normally associated with the data "type" (the SPF library type or CMS filetype). When the user edits a COBOL file, for example, the profile which was in effect the last time he finished editing a COBOL file is used. Additional profiles may be created by specifying a profile name on the edit file menu, or as an operand on the PROFILE command.

Listed below are the default mode settings when a new profile is generated (i.e., when the user edits a particular type of data for the first time, or specifies a new profile name on the edit menu). If the user already has 25 edit profiles, the least recently used profile is automatically deleted when a new profile is generated.

NUMBER	ON
AUTONUM	ON
STATS	ON
PRINT	OFF
CAPS	ON
NULLS	OFF
TABS	OFF
HEX	OFF
RECOVERY	OFF

Three of the mode settings will be automatically changed, if appropriate, whenever data is fetched for editing:

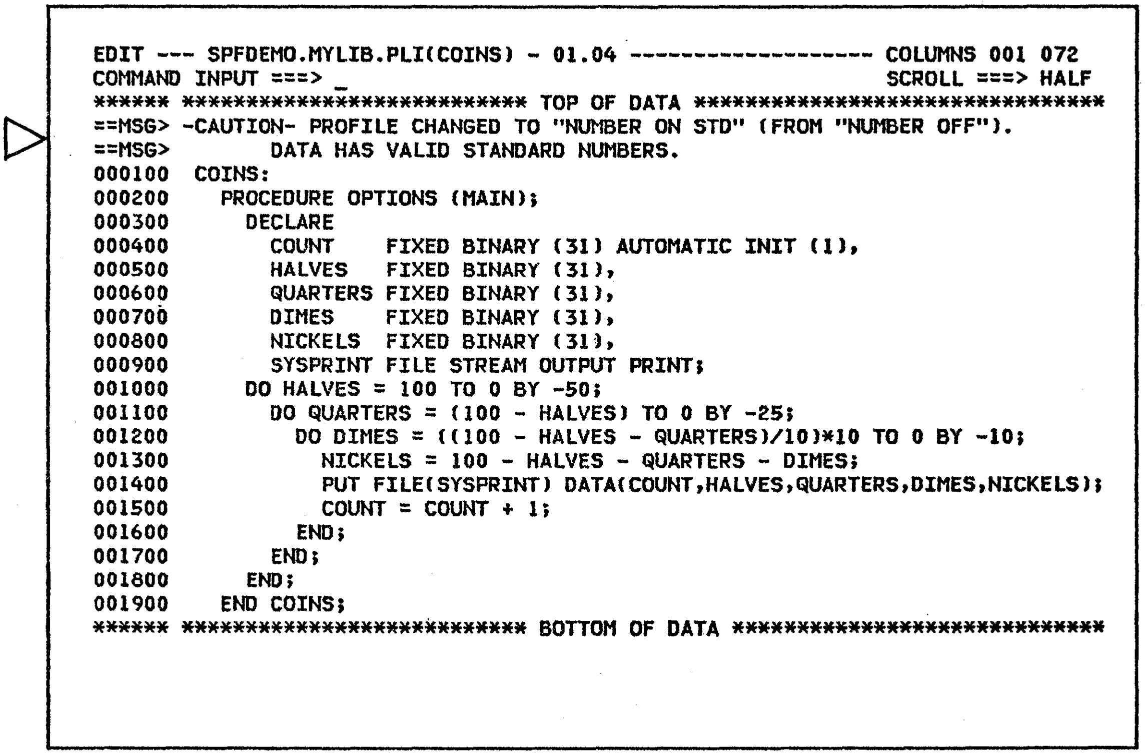
- NUMBER mode will be set on or off depending upon whether the data currently contains sequence numbers.

- CAPS mode will be set on or off depending upon whether the data currently contains lower case alphabetics.

- STATS mode will be set on if the member currently has SPF statistics. If STATS mode is already on and the member has no statistics, a caution message will be displayed but STATS mode will be left on. (STATS mode is ignored for CMS files.)

A caution message is displayed whenever edit automatically changes one of these mode settings. The message is displayed in the first two lines of the data area. An example is shown in Figure 25. A caution message is also displayed if edit discovers invalid (non-displayable) characters in the data.

Caution messages may be removed from display by using the RESET primary command or by deleting the individual message lines using the "D" line command. Caution messages are never saved as part of the data.



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------ COLUMNS 001 072
COMMAND INPUT ===> _                                        SCROLL ===> HALF
****** *************************** TOP OF DATA ******************************
==MSG> -CAUTION- PROFILE CHANGED TO "NUMBER ON STD" (FROM "NUMBER OFF").
==MSG>        DATA HAS VALID STANDARD NUMBERS.
000100 COINS:
000200   PROCEDURE OPTIONS (MAIN);
000300     DECLARE
000400       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500       HALVES   FIXED BINARY (31),
000600       QUARTERS FIXED BINARY (31),
000700       DIMES    FIXED BINARY (31),
000800       NICKELS  FIXED BINARY (31),
000900       SYSPRINT FILE STREAM OUTPUT PRINT;
001000     DO HALVES = 100 TO 0 BY -50;
001100       DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200         DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300           NICKELS = 100 - HALVES - QUARTERS - DIMES;
001400           PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001500           COUNT = COUNT + 1;
001600         END;
001700       END;
001800     END;
001900   END COINS;
****** *************************** BOTTOM OF DATA ****************************
```

Figure 25. Edit - caution message

Edit Termination

Normally, edit is terminated by pressing the End PF key, which causes the following actions:

1. If any changes have been made to the data:

   - The data is renumbered if both NUMBER mode and AUTONUM mode are on.

   - The data is automatically saved -- see description of SAVE command.

     Note: "Special" lines (MASK, TABS, COLS, BOUNDS, message lines, and profile display lines) are never saved as part of the data, and need not be deleted prior to issuing a SAVE command or pressing the End PF key.

   - The SPF statistics are updated (or generated if none previously existed) if an SPF library was specified and STATS mode is on.

   - A source listing of the data is recorded in the SPF list file for eventual printing if PRINT mode is on.

2. A return is then made to the previous menu (either the member selection list or the file menu). When return is to the member selection list, the member just edited will appear at the top of the list.

Note: The Return PF key, which is logically equivalent to repeated use of the End PF key, will also cause action (1) to occur.

The user may save the data without terminating edit (and without printing) via the SAVE command. The user may also terminate editing without saving (or printing) via the CANCEL command.

BASIC LINE COMMANDS

The following commands are described in this section:

COLS (columns)	M (move)
I (insert)	C (copy)
D (delete)	A (after)
R (repeat)	B (before)

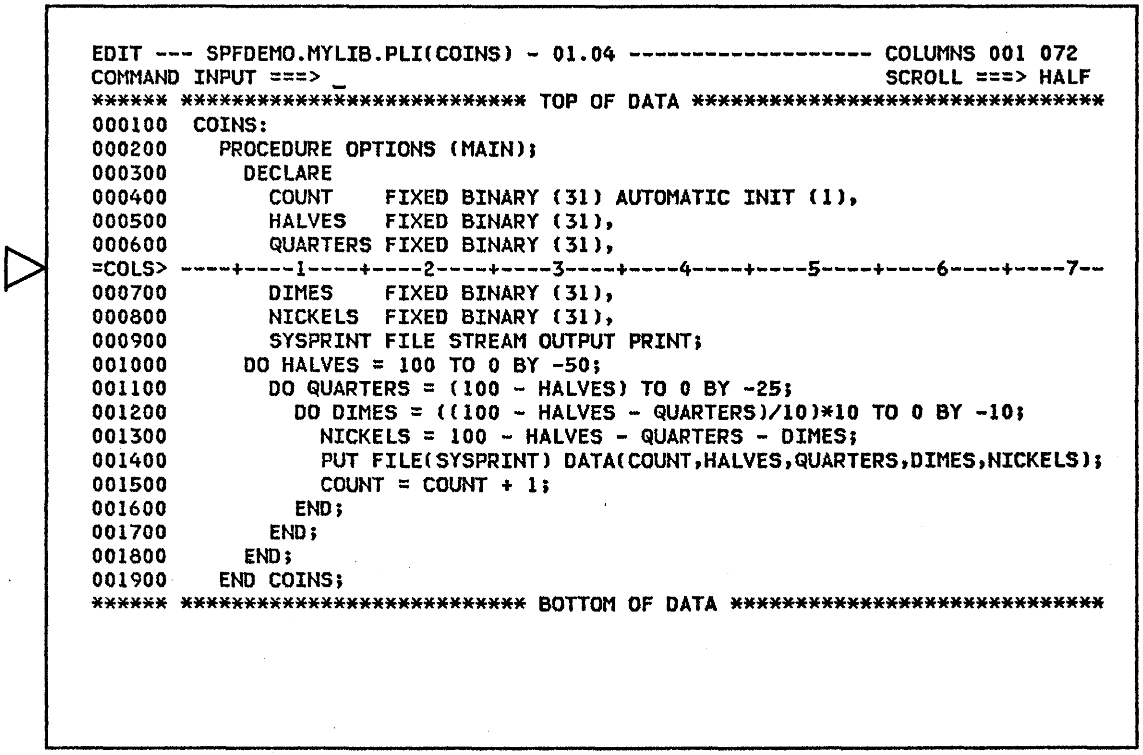
These are all "line" commands, entered by overtyping the 6-digit line number field on one or more lines.

Identifying Columns

A column identification line may be displayed by entering the characters "COLS" or "COL" in the line command area, overtyping the line number.

COLS - Display column identification line
Command abbreviation: COL

When the ENTER key is pressed, a special line is inserted at the designated position and the remaining lines are pushed down by one line position. An example of the COLS line is shown in Figure 26.



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------- COLUMNS 001 072
COMMAND INPUT ===> _                                          SCROLL ===> HALF
****** *************************** TOP OF DATA *******************************
000100  COINS:
000200    PROCEDURE OPTIONS (MAIN);
000300      DECLARE
000400        COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500        HALVES   FIXED BINARY (31),
000600        QUARTERS FIXED BINARY (31),
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000700        DIMES    FIXED BINARY (31),
000800        NICKELS  FIXED BINARY (31),
000900        SYSPRINT FILE STREAM OUTPUT PRINT;
001000      DO HALVES = 100 TO 0 BY -50;
001100        DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200          DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300            NICKELS = 100 - HALVES - QUARTERS - DIMES;
001400            PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001500            COUNT = COUNT + 1;
001600          END;
001700        END;
001800      END;
001900    END COINS;
****** *************************** BOTTOM OF DATA ****************************
```

Figure 26. Edit - column identification line

Note: Unlike browse, COLS is a line command, not a primary command (it may not be entered in the command input field on line 2).

The column identification line may be removed from display by entering a "D" in the line command area, or by entering a RESET primary command. The column identification line is never saved as part of the data.

Inserting and Deleting Lines

To insert a line, the single character "I" is entered in the line command area, overtyping the line number. When the ENTER key is pressed, a new line is inserted immediately following the line containing the "I". A number may follow the "I" to indicate that more than one line is to be inserted. For example, "I3" would cause three new lines to be inserted.

```
I   - Insert line
In  - Insert "n" lines
```

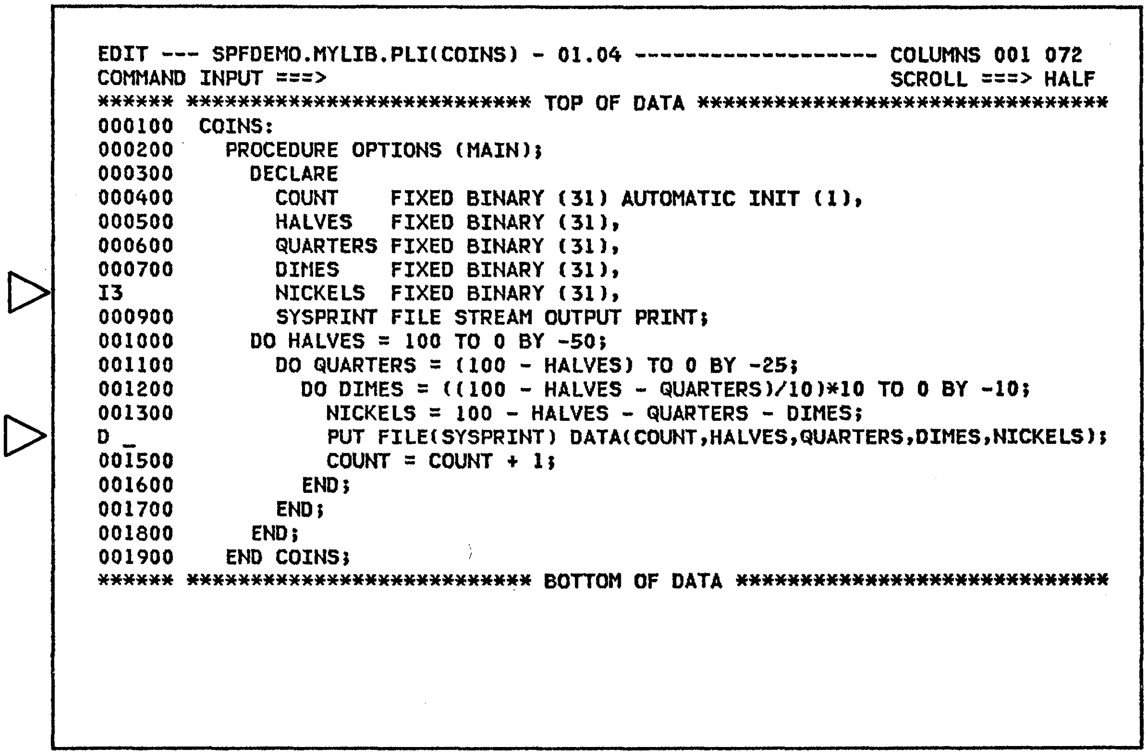
If any information is entered on an inserted line (even a blank character), the line becomes part of the source data and is assigned a line number the next time the ENTER key is pressed. If no information is entered on an inserted line, the line is automatically deleted the next time the ENTER key is pressed. If information is entered on the last (or only) inserted line and if the cursor is still in the data portion of that line when ENTER is pressed, another new line is automatically inserted following that line. This allows line after line of data to be generated in a "continuous insert" mode.

To delete a line, the single character "D" is entered in the line command area, overtyping the line number. A number may follow the "D" to indicate that more than one line is to be deleted. For example, "D99999" would cause all remaining lines to be deleted, starting with the line containing the "D99999".

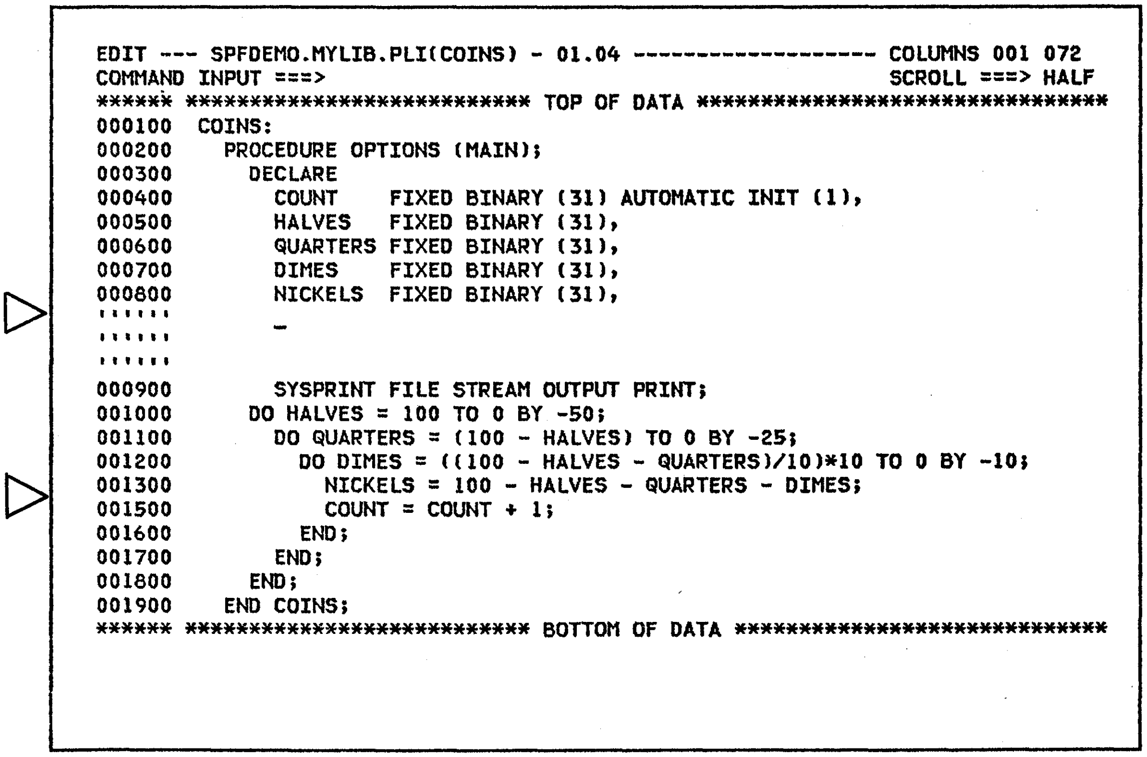
```
D   - Delete line
Dn  - Delete "n" lines
DD  - Delete block of lines
```

To delete a block of lines, the double character "DD" is entered in the line command area of the first and last lines to be deleted. The first and last line need not be on the same page; scrolling may be used between entering the first "DD" and the second "DD".

Figure 27 shows a before-and-after example of line insertion and deletion. Three new lines are inserted after line 800 and line 1400 is deleted.



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
****** *************************** TOP OF DATA *******************************
000100  COINS:
000200    PROCEDURE OPTIONS (MAIN);
000300      DECLARE
000400        COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500        HALVES   FIXED BINARY (31),
000600        QUARTERS FIXED BINARY (31),
000700        DIMES    FIXED BINARY (31),
I3            NICKELS  FIXED BINARY (31),
000900        SYSPRINT FILE STREAM OUTPUT PRINT;
001000      DO HALVES = 100 TO 0 BY -50;
001100        DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200          DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300            NICKELS = 100 - HALVES - QUARTERS - DIMES;
D _               PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001500            COUNT = COUNT + 1;
001600          END;
001700        END;
001800      END;
001900    END COINS;
****** *************************** BOTTOM OF DATA ****************************
```



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
****** *************************** TOP OF DATA *******************************
000100  COINS:
000200    PROCEDURE OPTIONS (MAIN);
000300      DECLARE
000400        COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500        HALVES   FIXED BINARY (31),
000600        QUARTERS FIXED BINARY (31),
000700        DIMES    FIXED BINARY (31),
000800        NICKELS  FIXED BINARY (31),
''''''        _
''''''
''''''
000900        SYSPRINT FILE STREAM OUTPUT PRINT;
001000      DO HALVES = 100 TO 0 BY -50;
001100        DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200          DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300            NICKELS = 100 - HALVES - QUARTERS - DIMES;
001500            COUNT = COUNT + 1;
001600          END;
001700        END;
001800      END;
001900    END COINS;
****** *************************** BOTTOM OF DATA ****************************
```

Figure 27. Edit - insert and delete lines example

Repeating Lines

To repeat a line, the single character "R" is entered in the line command area, overtyping the line number. When the ENTER key is pressed, a duplicate copy of the line is inserted immediately following the line containing the "R". A number may follow the "R" to indicate that the line is to be repeated more than one time. For example, "R5" would cause five identical copies of the line to be inserted.

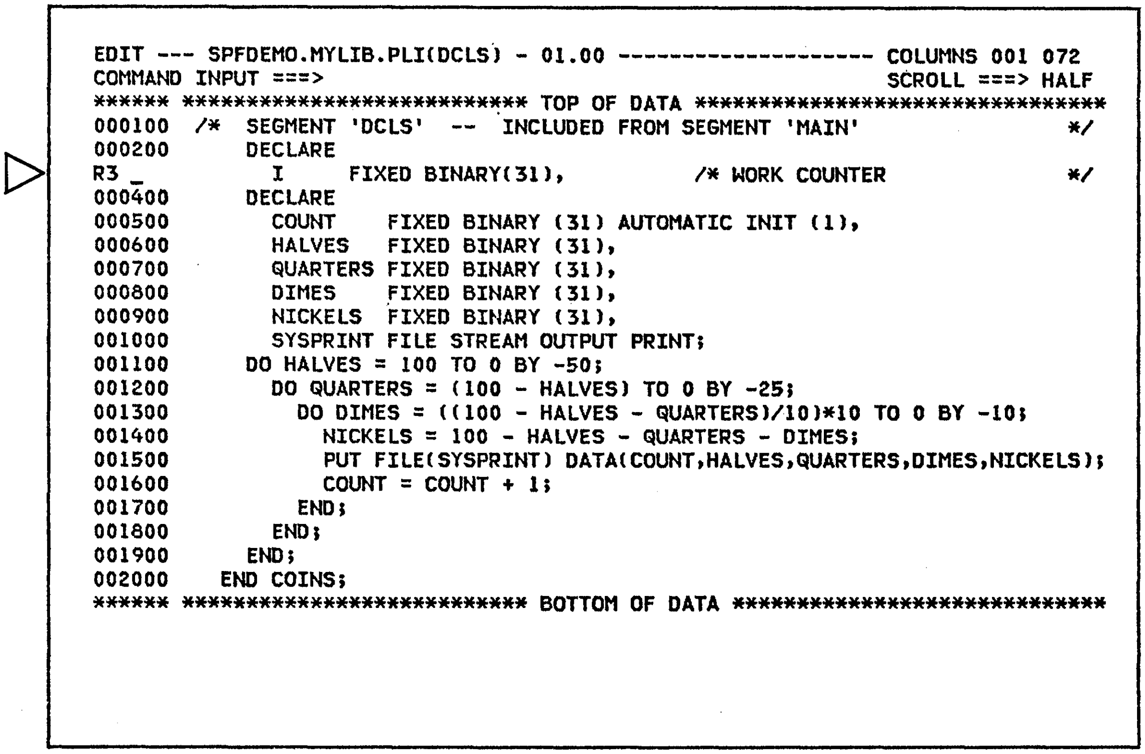
R	- Repeat line
Rn	- Repeat line "n" times
RR	- Repeat block of lines
RRn	- Repeat block of lines "n" times

To repeat a block of lines, the double character "RR" is entered in the line command area of the first and last lines to be repeated. The first and last lines need not be on the same page. A number may follow the "RR" on either the first or last lines (or both) to indicate that the block of lines is to be repeated more than one time.

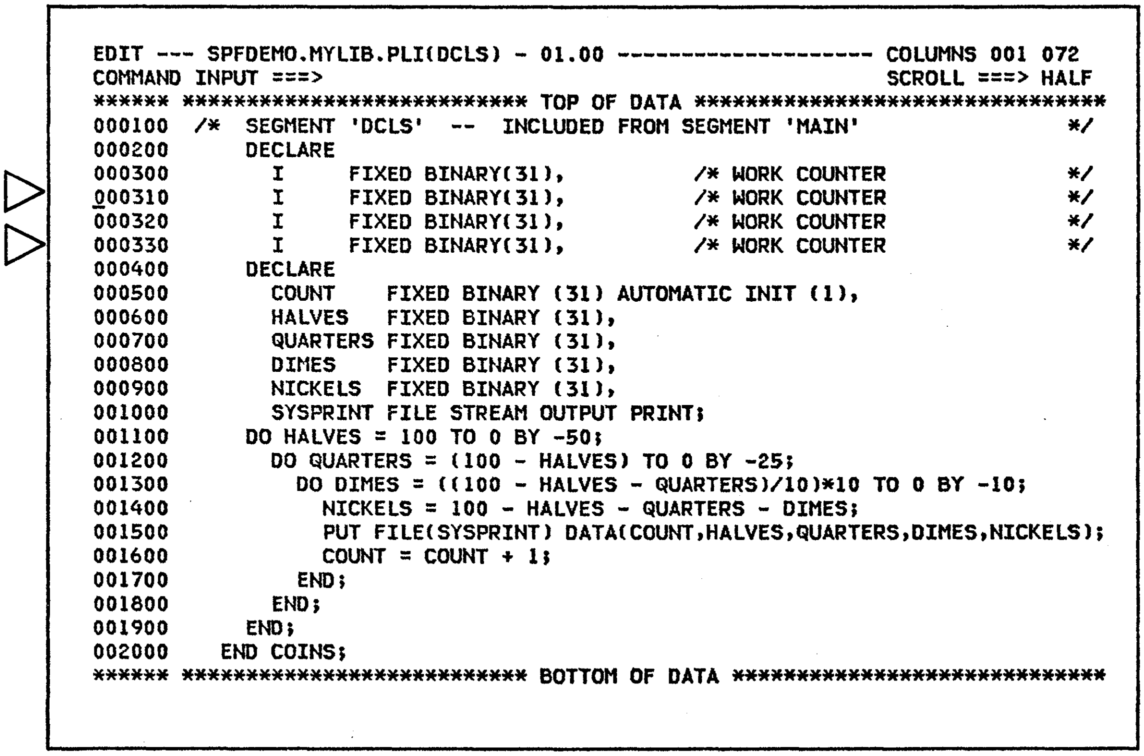
Suppose, for example, the following code is to be generated:

```
DECLARE
   I    FIXED BINARY(31),        /* WORK COUNTER */
   J    FIXED BINARY(31),        /* WORK COUNTER */
   K    FIXED BINARY(31),        /* WORK COUNTER */
   L    FIXED BINARY(31);        /* WORK COUNTER */
```

The easiest way to generate this code is to type the first two lines, repeat the second line three times, then move the cursor to each of the repeated lines and make the necessary changes. A before-and-after illustration of the repeat command is shown in Figure 28.



```
EDIT --- SPFDEMO.MYLIB.PLI(DCLS) - 01.00 -------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
****** *************************** TOP OF DATA *******************************
000100 /*  SEGMENT 'DCLS'  --  INCLUDED FROM SEGMENT 'MAIN'                   */
000200     DECLARE
R3 _         I      FIXED BINARY(31),         /* WORK COUNTER              */
000400     DECLARE
000500       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000600       HALVES   FIXED BINARY (31),
000700       QUARTERS FIXED BINARY (31),
000800       DIMES    FIXED BINARY (31),
000900       NICKELS  FIXED BINARY (31),
001000       SYSPRINT FILE STREAM OUTPUT PRINT;
001100     DO HALVES = 100 TO 0 BY -50;
001200       DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001300         DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001400           NICKELS = 100 - HALVES - QUARTERS - DIMES;
001500           PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001600           COUNT = COUNT + 1;
001700         END;
001800       END;
001900     END;
002000   END COINS;
****** *************************** BOTTOM OF DATA ****************************
```



```
EDIT --- SPFDEMO.MYLIB.PLI(DCLS) - 01.00 -------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
****** *************************** TOP OF DATA *******************************
000100 /*  SEGMENT 'DCLS'  --  INCLUDED FROM SEGMENT 'MAIN'                   */
000200     DECLARE
000300       I      FIXED BINARY(31),         /* WORK COUNTER              */
000310       I      FIXED BINARY(31),         /* WORK COUNTER              */
000320       I      FIXED BINARY(31),         /* WORK COUNTER              */
000330       I      FIXED BINARY(31),         /* WORK COUNTER              */
000400     DECLARE
000500       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000600       HALVES   FIXED BINARY (31),
000700       QUARTERS FIXED BINARY (31),
000800       DIMES    FIXED BINARY (31),
000900       NICKELS  FIXED BINARY (31),
001000       SYSPRINT FILE STREAM OUTPUT PRINT;
001100     DO HALVES = 100 TO 0 BY -50;
001200       DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001300         DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001400           NICKELS = 100 - HALVES - QUARTERS - DIMES;
001500           PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001600           COUNT = COUNT + 1;
001700         END;
001800       END;
001900     END;
002000   END COINS;
****** *************************** BOTTOM OF DATA ****************************
```

Figure 28. Edit - repeat line example

Moving and Copying Lines

To move a line, the single character "M" is entered in the line command area of the line to be moved, and the character "A" (after) is entered in the line command area of another line, indicating the destination of the move. When the ENTER key is pressed, the line containing the "M" is moved immediately following the line containing the "A". Alternatively, the destination may be indicated with a "B" (before) line command, in which case the line containing the "M" is inserted immediately preceding the line containing the "B".

M	- Move line
Mn	- Move "n" lines
MM	- Move block of lines

C	- Copy line
Cn	- Copy "n" lines
CC	- Copy block of lines

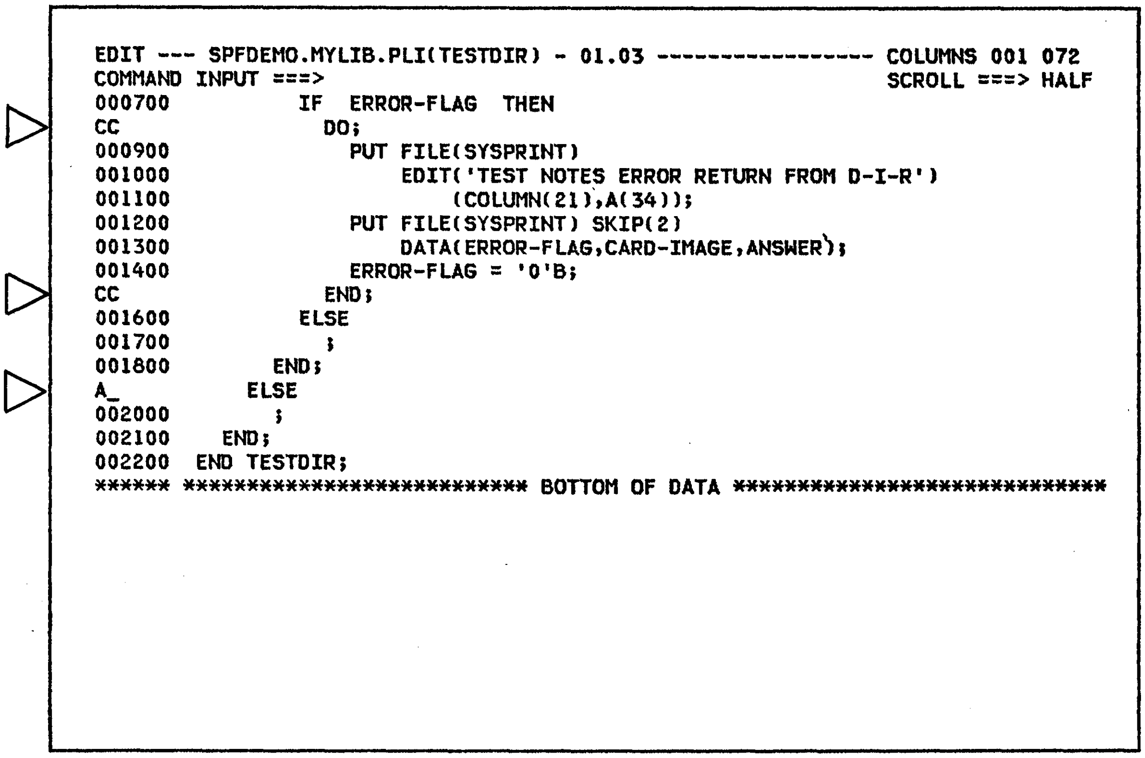
A	- After this line
An	- After this line and repeat "n" times
B	- Before this line
Bn	- Before this line and repeat "n" times

A number may follow the "M" to indicate that more than one line is to be moved. A block of lines to be moved is indicated by entering the double character "MM" on the first and last lines to be moved. The first and last lines to be moved, and the destination line may all be on separate pages.

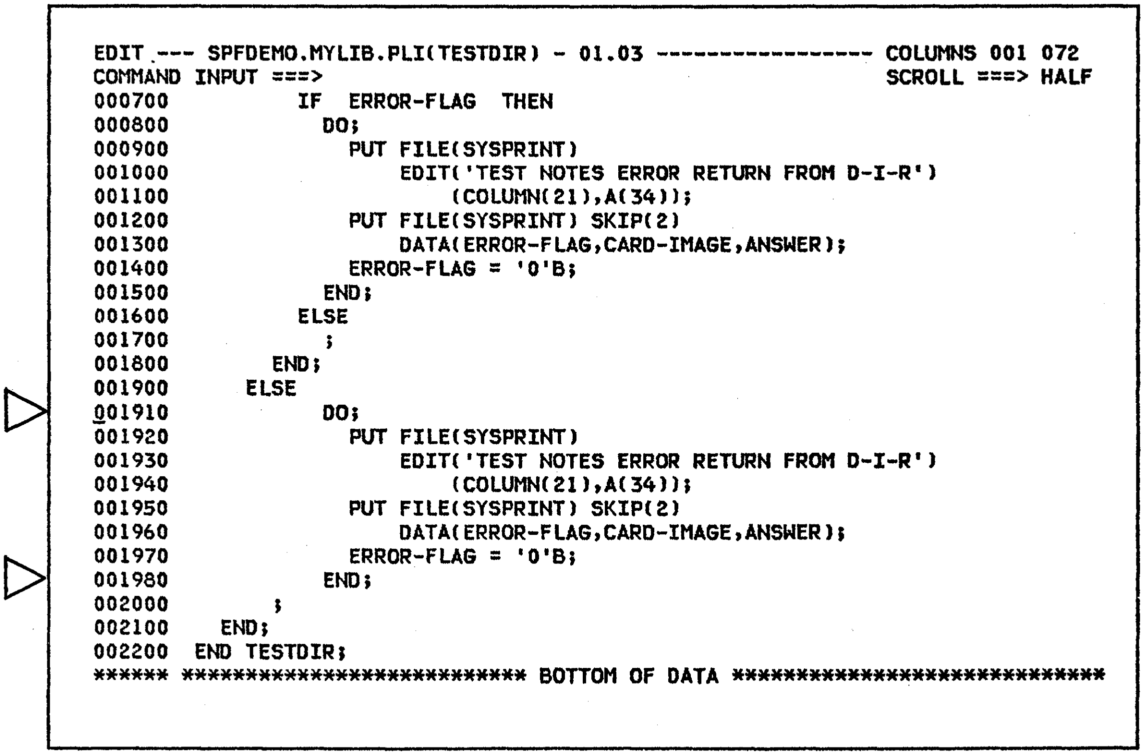
A number may also follow the "A" or "B" line command to cause the moved line(s) to be repeated the designated number of times at the point of destination.

The procedure for copying lines is the same as for moving lines, except that a "C" or "CC" line command is used in place of the "M" or "MM". The copy operation leaves the original line(s) in place, and makes a duplicate copy at the indicated destination.

Figure 29 shows a before-and-after example of copying lines. The block of lines starting at line 800 through line 1500 is copied to follow line 1900.



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                             SCROLL ===> HALF
000700          IF  ERROR-FLAG  THEN
CC                DO;
000900              PUT FILE(SYSPRINT)
001000                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001100                      (COLUMN(21),A(34));
001200              PUT FILE(SYSPRINT) SKIP(2)
001300                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001400              ERROR-FLAG = '0'B;
CC                END;
001600          ELSE
001700            ;
001800        END;
A_          ELSE
002000        ;
002100    END;
002200  END TESTDIR;
****** *************************** BOTTOM OF DATA ****************************
```



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                             SCROLL ===> HALF
000700          IF  ERROR-FLAG  THEN
000800            DO;
000900              PUT FILE(SYSPRINT)
001000                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001100                      (COLUMN(21),A(34));
001200              PUT FILE(SYSPRINT) SKIP(2)
001300                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001400              ERROR-FLAG = '0'B;
001500            END;
001600          ELSE
001700            ;
001800        END;
001900      ELSE
001910            DO;
001920              PUT FILE(SYSPRINT)
001930                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001940                      (COLUMN(21),A(34));
001950              PUT FILE(SYSPRINT) SKIP(2)
001960                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001970              ERROR-FLAG = '0'B;
001980            END;
002000        ;
002100    END;
002200  END TESTDIR;
****** *************************** BOTTOM OF DATA ****************************
```

Figure 29. Edit - copy lines example

Rules for Entering Line Commands

The following rules apply to all edit line commands.

Several line commands as well as data modifications may be typed before pressing the ENTER key. Error messages will be displayed in cases where multiple line commands are ambiguous. If an erroneous or unwanted line command is entered, it may be undone by simply blanking it out or by using the RESET primary command.

In most cases, only the first one or two characters of the line number need be overtyped to enter a line command. For example, if line 700 is to be repeated, a single "R" may be typed:

```
000600
R00700
000800
```

In some cases, however, typing a single character may be ambiguous. In the following example, it is unclear whether the user has typed a single "R" to repeat line 31700, or "R3" to repeat the line three times.

```
031600
R31700
031800
```

In such cases, edit assumes that the user has not entered a number following the line command. If the user meant to repeat the line three times, he may use any of the following procedures.

1. Leave the cursor on the character immediately following the "R3": R31700

2. Type one or more blanks following the "R3": R3 700

3. Type "R3" and then press the ERASE EOF key to clear the rest of the line number field (or press ERASE EOF before typing "R3").

Three line commands may be entered on the "TOP OF DATA" line by overtyping the asterisks which normally appear in its line number field. They are:

I or In - to insert one or more lines ahead of the rest of the data.

A or An - to move or copy lines ahead of the rest of the data.

TE or TEn - to enter one or more lines of text ahead of the rest of the data.

One line command may be entered on the "BOTTOM OF DATA" line by overtyping the asterisks, namely:

B or Bn - to move or copy lines following the rest of the data.

Hints on use of cursor keys: Use the NEW LINE key to move the cursor to the line command area of the next line. If the cursor is in a line command area, use the TAB FORWARD key to space over to the data field.

Note: When a line command or data modification is typed and the ENTER key is pressed, edit will frequently reposition the cursor to another line number or to another location within the data. Edit attempts to guess the most appropriate position for the cursor, based on the previous user action. To assist the user in locating the cursor, edit intensifies the line number field on the line that contains the cursor.

GENERAL COMMANDS

The following commands are described in this section:

LOCATE
NUMBER
RENUM
UNNUM
AUTONUM
STATS
PRINT
RECOVERY
RESET
SUBMIT
SAVE
CANCEL

These are all "primary" commands, entered in the command input field on line 2.

Locating Lines

The LOCATE command performs up or down scrolling, as required, to a specified line number. If the specified line is not found, scrolling is to the line which precedes the specified line number. Format:

LOCATE line-number
Command abbreviations: LOC, L

The "line-number" operand is a numeric value of up to eight digits (leading zeros need not be typed). If the operand contains six or fewer digits, it refers to the line-number fields displayed at the left of each line. (For numbered data, these fields reflect the sequence numbers in the data records. For unnumbered data, these fields show relative line numbers.) If the line-number operand contains eight digits, it refers directly to the sequence numbers in the data records, including the "modification flag" for SPF libraries (see below).

Sequence Number Generation and Control

When operating in NUMBER mode, edit automatically generates sequence numbers on new lines that are created by insert or copy operations, and automatically renumbers the data when it is saved (provided AUTONUM mode is also in effect).

Sequence Number Format

Sequence numbers may be generated in the "standard" sequence field, the COBOL sequence field, or both. The standard sequence field is either the last eight characters (for fixed length records) or the first eight characters (for variable length records), regardless of the programming language. The COBOL sequence field is always the first six characters of the line, and is valid only for fixed length records.

For members of SPF libraries, the format of "standard" sequence numbers is dependent on whether SPF statistics are being generated. If STATS mode is in effect, standard sequence numbers are six digits in length, followed by a two-digit "modification flag". The flag reflects the modification level of the member when the line was created or last changed. If, for example, a sequence number field contains 00040002, the line was added or last changed at modification level 02. The sequence number is 000400. When a new member is created with SPF statistics, the modification flags are all set to 00, since the modification level is initialized to zero.

If STATS mode is off, or if a CMS file is being edited, standard sequence numbers are eight digits in length (right justified within the field). COBOL sequence numbers are always six digits in length, and are unaffected by the setting of STATS mode.

Sequence numbers normally start at 100 and are incremented by 100. When inserting lines, edit will drop down to the tens or units position and, if necessary, renumber one or more succeeding lines to keep the sequence numbers in order.

Initialization of NUMBER Mode

As data is fetched for editing, it is examined to determine whether or not it contains sequence numbers. The standard sequence field is always examined. The COBOL sequence field is also examined if the SPF library type or CMS filetype is COBOL.

If all lines contain numeric characters in either the standard or COBOL sequence field positions (or both), and if the numbers are in ascending order, edit assumes the data is numbered and turns on NUMBER mode. Otherwise, edit turns off NUMBER mode.

If the initial setting of NUMBER mode differs from the previous setting in the profile, a message will be displayed indicating that edit has changed the mode. For new files or members, the initial setting of NUMBER mode is unchanged from whatever is specified in the profile. The default for a new profile is NUMBER mode ON for standard sequence fields, and for COBOL fields if the type is COBOL.

Controlling Sequence Numbering

NUMBER mode may be turned on or off by using the NUMBER command. Format:

```
   NUMBER  [ON/OFF]  [STD]  [COBOL]

   Command abbreviation:  NUM
   Keyword abbreviation:  COB (for COBOL)
```

The operands may be typed in any order. If the ON/OFF operand is omitted, ON is the default. The STD and COBOL operands are interpreted only when NUMBER mode is turned on. If STD typed, the data will be numbered in the standard sequence field. If COBOL is typed, the data will be numbered in the COBOL field. If both operands are typed, the data will be numbered in both fields. If both operands are omitted, the default is STD unless NUMBER mode was already in effect, in which case the data will be numbered in whichever field(s) were previously being numbered.

Note: If both STD and COBOL numbers are being generated, the STD number is determined and then used as the COBOL number. This can result in COBOL numbers which are out of sequence if the COBOL and STD fields are not synchronized. Use the RENUM command (see below) to force synchronization.

When NUMBER mode is turned on, the NUMBER command verifies that all lines have valid numbers that are in ascending sequence. It renumbers any lines that are either unnumbered or out of sequence, but does not otherwise alter existing sequence numbers.

Three additional commands, RENUM, UNNUM, and AUTONUM, are available to assist with number generation and control.

The RENUM command immediately renumbers all lines, starting with number 100 and incrementing by 100, and also turns on NUMBER mode. Format:

```
RENUM  [STD]  [COBOL]

Command abbreviation:  REN
Keyword abbreviation:  COB (for COBOL)
```

The meaning of the STD and COBOL operands is the same as for the NUMBER command.

The UNNUM command sets all sequence fields to blanks and also turns off NUMBER mode. It is valid only when NUMBER mode is currently on. The fields to be blanked are either the standard sequence field, the COBOL sequence field, or both, depending what was previously being numbered prior to issuing the UNNUM command. Format:

```
UNNUM

Command abbreviation:  UNN
```

The UNNUM command uses no operands.

The AUTONUM command controls the setting of AUTONUM mode. When this mode is on, the data is automatically renumbered whenever it is saved, provided NUMBER mode is also on at that time.

AUTONUM mode may be turned on or off by using the AUTONUM command. Format:

```
AUTONUM  [ON/OFF]
```

If the operand is omitted, ON is the default.

Statistics Generation

Generation of SPF statistics for a member of a an SPF library is controlled by STATS mode. STATS mode is ignored for CMS files.

If STATS mode is on when the member is saved, the statistics will be updated (or created if the member did not previously have statistics). If STATS mode is off when the member is saved, no statistics are stored and any previous statistics are destroyed. See Appendix B for more information about the maintenance of SPF statistics.

STATS mode may be turned on or off by using the STATS command. Format:

```
STATS  [ON/OFF]
```

If the operand is omitted, ON is the default.

Whenever a member is fetched for editing, SPF will check the setting of STATS mode. If STATS mode is off and the member has statistics, SPF will automatically turn on STATS mode and display a message indicating that the mode has been changed. If STATS mode is on and the member has no statistics, a message will be displayed but the mode will remain unchanged.

When a new member is created, the initial setting of STATS mode is unchanged from whatever is in the profile. The default for a new profile is STATS mode ON.

The generation of SPF statistics also affects the format of sequence numbers, as discussed under "Sequence Number Generation and Control".

Automatic Source Listings

The automatic generation of source listings is controlled by PRINT mode. If this mode is on when edit is terminated via the End or Return PF key, and if the data has been changed, a source listing of the data will be recorded in the SPF list file for eventual printing.

PRINT mode may be turned on or off by using the PRINT command. Format:

```
PRINT  [ON/OFF]
```

If the operand is omitted, ON is the default.

Recovery Control

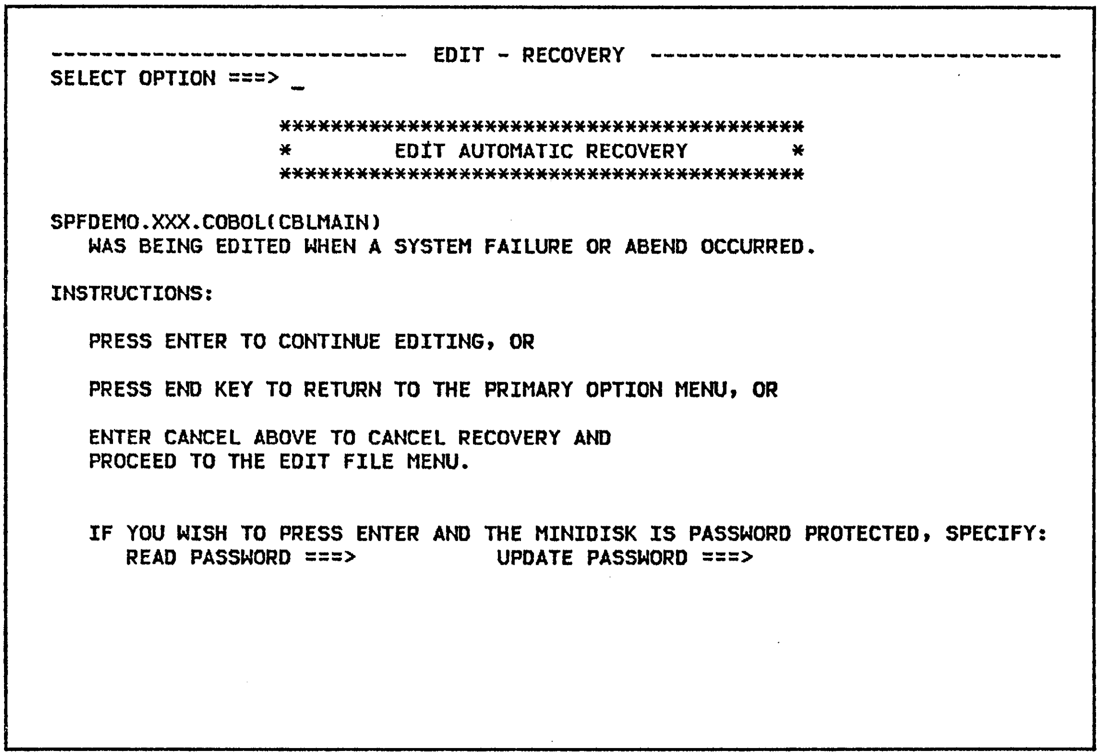
When edit is operating in RECOVERY mode, an audit trail of user interactions is automatically recorded in an SPF-controlled file. Following a system failure, the recovery file may be used to replay the edit session. When edit is selected from the primary option menu, a special menu is displayed indicating that recovery is available (Figure 30). As directed on the menu, the user may proceed with recovery, cancel recovery, or press the End PF key to return to the primary option menu.

RECOVERY mode may be turned on or off by using the RECOVERY command. Format:

```
RECOVERY  [ON/OFF]

Command abbreviation:  REC
```

If the operand is omitted, ON is the default.

Operating with RECOVERY mode off may result in improved response time due to the elimination of I/O operations to maintain the recovery file. If RECOVERY mode is on, the first change to the data causes a checkpoint of the data to be written. This may increase response time for the interaction, especially if a large member or file is being edited.



```
---------------------------- EDIT - RECOVERY -------------------------------
SELECT OPTION ===> _

                  *****************************************
                  *         EDIT AUTOMATIC RECOVERY        *
                  *****************************************

SPFDEMO.XXX.COBOL(CBLMAIN)
   WAS BEING EDITED WHEN A SYSTEM FAILURE OR ABEND OCCURRED.

INSTRUCTIONS:

   PRESS ENTER TO CONTINUE EDITING, OR

   PRESS END KEY TO RETURN TO THE PRIMARY OPTION MENU, OR

   ENTER CANCEL ABOVE TO CANCEL RECOVERY AND
   PROCEED TO THE EDIT FILE MENU.


   IF YOU WISH TO PRESS ENTER AND THE MINIDISK IS PASSWORD PROTECTED, SPECIFY:
      READ PASSWORD ===>           UPDATE PASSWORD ===>
```

Figure 30. Edit - recovery menu

General Reset

Edit will display warning messages in the data portion of the display if a mode setting has been automatically changed by edit after examination of the data, or if the data contains invalid characters. Edit also displays line messages to show the current setting of edit modes (when the PROFILE command is entered). Display of messages and other special conditions may be reset by using the RESET command. Format:

RESET
Command abbreviation: RES

No operands are used for this command.

The RESET command will: terminate display of all message lines and other special lines (COLS, MASK, TABS, and BOUNDS), reset line numbers to normal after they were replaced with "==CHG>" or "==ERR>" messages (see CHANGE and shift commands), blank out erroneous or unwanted line commands, and redisplay all excluded lines. The RESET command does not change any of the edit modes, nor alter the data.

Note: RESET scans every line of data for conditions to be reset. Use of the "D" line command to delete one or more special lines may result in faster response time.

Submit to Job Stream

The SUBMIT command may be used to submit the data being edited (entire member or file) to the job stream of another VM machine. Format:

```
SUBMIT  machine-id
---------------------------------
Command abbreviation:  SUB
```

The operand is required to specify the VM machine to which the data is being submitted. It may specify a CMS batch machine, or a VS or DOS machine.

Note: SPF does not supply a job statement when the edit SUBMIT command is invoked. The user may supply job statement(s) as part of the data being submitted.

Save and Cancel

The SAVE command may be used to save the edited data without terminating edit, and the CANCEL command may be used to terminate edit without saving the data. Formats:

```
SAVE
```

```
CANCEL
---------------------------------
Command abbreviation:  CAN
```

No operands are used for these commands.

Note: As a general rule, use of the SAVE command is not necessary if RECOVERY mode is on.

The SAVE command writes the data back to the same file or member from where it was fetched, except when a concatenated sequence of SPF libraries has been specified. In this case, the data is saved in the first library of the concatenation sequence regardless of which library it came from. For a sequential file, the entire file is rewritten. For an SPF library, MACLIB, or TXTLIB, the member is rewritten with the same member name. The statistics for a member of an SPF library are automatically updated (provided STATS mode is in effect).

The data is automatically renumbered prior to saving, provided both NUMBER mode and AUTONUM mode are in effect.

If SAVE cannot successfully rewrite the data due to I/O errors or insufficient space on the minidisk, a message is displayed in the upper right-hand corner of the screen, accompanied by an audible alarm (if installed). The user may then attempt to save the data in another file by taking the following steps:

1. Enter a CREATE or REPLACE command with no operand on the primary command line.

2. Enter "C99999" or "M99999" on the first data line, indicating that all lines are to be copied or moved. Then press the ENTER key.

3. Fill in the file or library (and member) name on the CREATE or REPLACE menu, and press the ENTER key.

See CREATE and REPLACE commands for further information.

Alternatively, the user may wish to enter split screen mode and attempt to fix the problem by using appropriate utilities, e.g., compress or delete, and then retry the save from the edit screen.

The normal way to terminate edit is by pressing the End or Return PF key, which causes the following actions.

1. A SAVE is issued if any changes have been made since the data was last saved. (The above discussion of I/O errors also pertains to an automatic SAVE when the End or Return key is pressed.)

2. If any saves have occured, the data is recorded in the SPF list file (provided PRINT mode is in effect).

3. Editing is then terminated.

Note that neither the SAVE nor CANCEL command causes automatic recording in the SPF list file, regardless of the setting of PRINT mode.

FINDING AND CHANGING CHARACTER STRINGS

The following commands are described in this section:

```
FIND
CHANGE
```

These are both "primary" commands, entered in the command input field on line 2.

The FIND command is used to find one or more occurrances of a specified character string, and the CHANGE command is used to change one or more occurrences to another character string. Both commands allow special forms of character strings and several optional parameters to control:

- Starting point and direction of search
- Special conditions to control character string "match"
- Range and column limitations on extent of search

The command summary, which follows, describes the basic forms of FIND and CHANGE which are most frequently needed. The remainder of this section provides detailed information about the various special features and options.

Command Summary

The basic format for FIND and CHANGE is:

FIND string-1 [ALL]
Command abbreviation: F

CHANGE string-1 string-2 [ALL]
Command abbreviations: CHG, C

The operands may be separated by blanks or commas. The ALL operand is optional, and may precede or follow either string.

As a general rule, the strings are typed without delimiters. Examples:

```
FIND  XYZ
CHANGE  ALL  ABC  IJKL
```

to find the next occurrence of XYZ or to change all occurrences of ABC to IJKL.

Strings may be delimited with either apostrophes (') or quotation marks ("). Delimiters must be used if a string contains imbedded blanks or commas, or if a string is the same as a command keyword. Examples:

```
FIND  'EVERY ONE'
CHANGE  'EVERY ONE' 'ALL'
```

to find the next occurrence of "EVERY ONE" or to change the next occurrence of "EVERY ONE" to "ALL".

If the ALL operand is omitted, the FIND or CHANGE command searches for the next occurrence of string-1 starting at the current cursor location. (If the cursor is not in the data area of the screen, scanning starts at the beginning of the first line which is currently being displayed.) Scrolling is performed, if necessary, to bring the string into view. The cursor is positioned under the string (for a CHANGE command, it is positioned at the end of the changed string) and a verification message is displayed in the upper right-hand corner of the screen. The Find or Change PF key may be used to find or change each successive occurrence of the string.

If the string is not found between the current cursor location and the end of data, a "BOTTOM OF DATA REACHED" message is displayed and the audible alarm is sounded (if installed). The Find or Change PF key may then be used to wrap to the top of data and continue searching. If the string is not found anywhere in the data a "NO xxxxx FOUND" message is displayed.

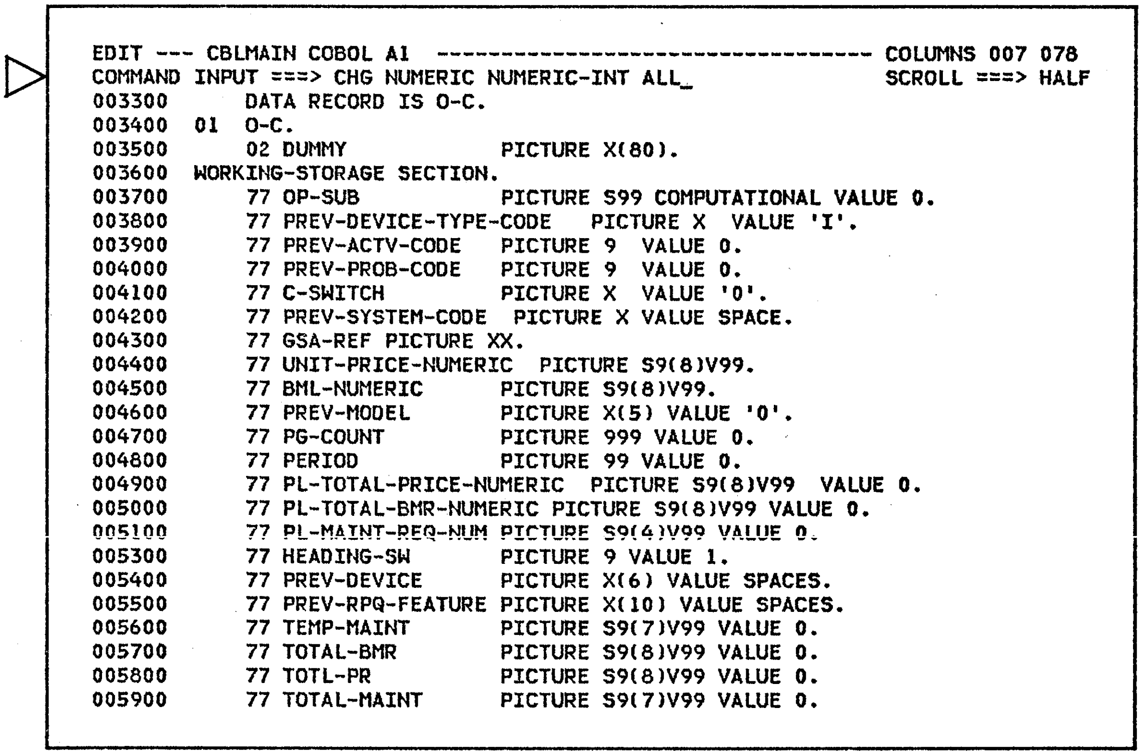
If the ALL operand is typed, the FIND or CHANGE command searches for all occurrences of string-1 starting at the top of the data, and positions the cursor under the first occurrence. The verification message indicates the number of occurrences found. For a CHANGE command, each line on which a change occurred is indicated with a "==CHG>" message in the line number field.

If the two strings specified for a CHANGE command are not the same length, automatic shifting is performed by expanding or collapsing multiple blank characters to the right of the substitution. In no case is data lost. If insufficient blanks exist for right-shifting, the original string remains unchanged. This condition is indicated with a "==ERR>" message in the line number field.

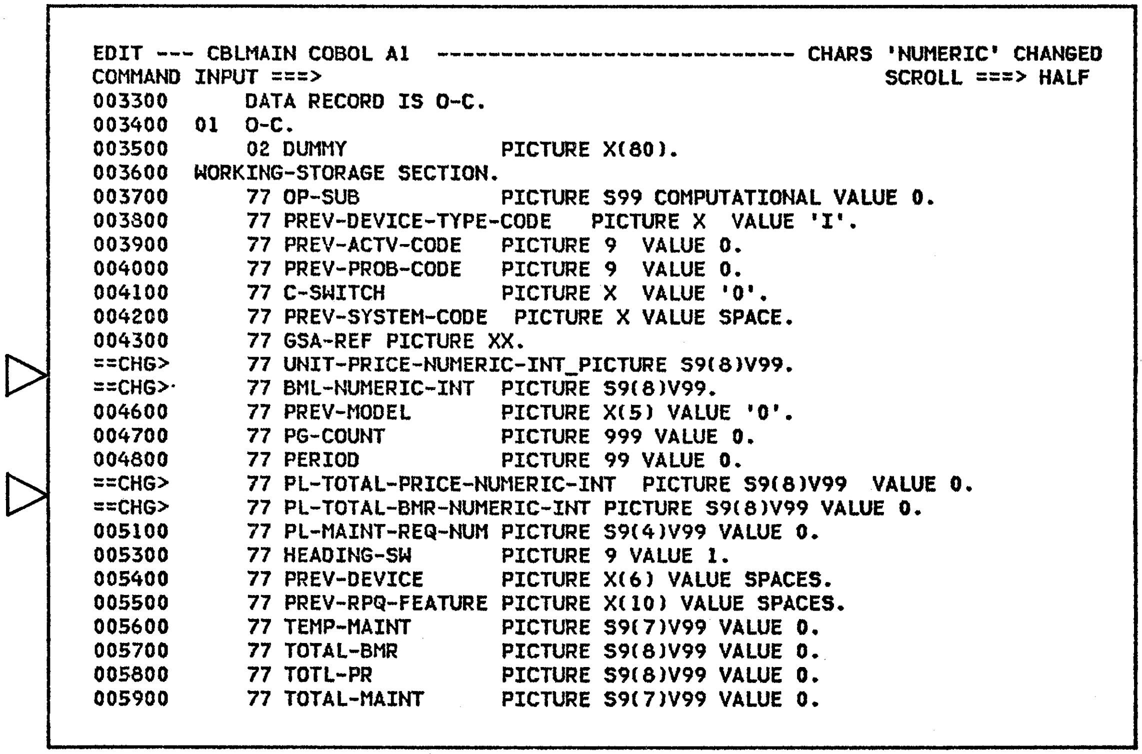
Following a CHANGE command, the ==CHG> and ==ERR> line messages may be reset to normal via the RESET command, or by overtyping the line number or data on those lines.

The Find and Change PF keys may be used together to find each occurrence of a character string, examine it, and then either change it (by pressing the Change PF key), or go on to the next occurrence (by pressing the Find PF key). To do this, start by typing a CHANGE command on line 2 but then, rather than pressing ENTER, press the Find PF key. This will position the cursor to string-1 without changing it. Then press Change (to change it to string-2) or Find (to get to the next occurrence of string-1).

Figure 31 shows a before-and-after example of the CHANGE comand. All occurrences of the character string NUMERIC are replaced with the string NUMERIC-INT. The cursor is repositioned to the end of the first occurrence, and a ==CHG> message is displayed at the beginning of each changed line.



```
EDIT --- CBLMAIN COBOL A1  ---------------------------------- COLUMNS 007 078
COMMAND INPUT ===> CHG NUMERIC NUMERIC-INT ALL_              SCROLL ===> HALF
003300       DATA RECORD IS O-C.
003400   01  O-C.
003500       02 DUMMY            PICTURE X(80).
003600   WORKING-STORAGE SECTION.
003700       77 OP-SUB           PICTURE S99 COMPUTATIONAL VALUE 0.
003800       77 PREV-DEVICE-TYPE-CODE   PICTURE X  VALUE 'I'.
003900       77 PREV-ACTV-CODE   PICTURE 9  VALUE 0.
004000       77 PREV-PROB-CODE   PICTURE 9  VALUE 0.
004100       77 C-SWITCH         PICTURE X  VALUE '0'.
004200       77 PREV-SYSTEM-CODE  PICTURE X VALUE SPACE.
004300       77 GSA-REF PICTURE XX.
004400       77 UNIT-PRICE-NUMERIC  PICTURE S9(8)V99.
004500       77 BML-NUMERIC      PICTURE S9(8)V99.
004600       77 PREV-MODEL       PICTURE X(5) VALUE '0'.
004700       77 PG-COUNT         PICTURE 999 VALUE 0.
004800       77 PERIOD           PICTURE 99 VALUE 0.
004900       77 PL-TOTAL-PRICE-NUMERIC  PICTURE S9(8)V99  VALUE 0.
005000       77 PL-TOTAL-BMR-NUMERIC PICTURE S9(8)V99 VALUE 0.
005100       77 PL-MAINT-REQ-NUM PICTURE S9(4)V99 VALUE 0.
005300       77 HEADING-SW       PICTURE 9 VALUE 1.
005400       77 PREV-DEVICE      PICTURE X(6) VALUE SPACES.
005500       77 PREV-RPQ-FEATURE PICTURE X(10) VALUE SPACES.
005600       77 TEMP-MAINT       PICTURE S9(7)V99 VALUE 0.
005700       77 TOTAL-BMR        PICTURE S9(8)V99 VALUE 0.
005800       77 TOTL-PR          PICTURE S9(8)V99 VALUE 0.
005900       77 TOTAL-MAINT      PICTURE S9(7)V99 VALUE 0.
```



```
EDIT --- CBLMAIN COBOL A1  ---------------------------- CHARS 'NUMERIC' CHANGED
COMMAND INPUT ===>                                           SCROLL ===> HALF
003300       DATA RECORD IS O-C.
003400   01  O-C.
003500       02 DUMMY            PICTURE X(80).
003600   WORKING-STORAGE SECTION.
003700       77 OP-SUB           PICTURE S99 COMPUTATIONAL VALUE 0.
003800       77 PREV-DEVICE-TYPE-CODE   PICTURE X  VALUE 'I'.
003900       77 PREV-ACTV-CODE   PICTURE 9  VALUE 0.
004000       77 PREV-PROB-CODE   PICTURE 9  VALUE 0.
004100       77 C-SWITCH         PICTURE X  VALUE '0'.
004200       77 PREV-SYSTEM-CODE  PICTURE X VALUE SPACE.
004300       77 GSA-REF PICTURE XX.
==CHG>       77 UNIT-PRICE-NUMERIC-INT_PICTURE S9(8)V99.
==CHG>       77 BML-NUMERIC-INT  PICTURE S9(8)V99.
004600       77 PREV-MODEL       PICTURE X(5) VALUE '0'.
004700       77 PG-COUNT         PICTURE 999 VALUE 0.
004800       77 PERIOD           PICTURE 99 VALUE 0.
==CHG>       77 PL-TOTAL-PRICE-NUMERIC-INT  PICTURE S9(8)V99  VALUE 0.
==CHG>       77 PL-TOTAL-BMR-NUMERIC-INT PICTURE S9(8)V99 VALUE 0.
005100       77 PL-MAINT-REQ-NUM PICTURE S9(4)V99 VALUE 0.
005300       77 HEADING-SW       PICTURE 9 VALUE 1.
005400       77 PREV-DEVICE      PICTURE X(6) VALUE SPACES.
005500       77 PREV-RPQ-FEATURE PICTURE X(10) VALUE SPACES.
005600       77 TEMP-MAINT       PICTURE S9(7)V99 VALUE 0.
005700       77 TOTAL-BMR        PICTURE S9(8)V99 VALUE 0.
005800       77 TOTL-PR          PICTURE S9(8)V99 VALUE 0.
005900       77 TOTAL-MAINT      PICTURE S9(7)V99 VALUE 0.
```

Figure 31. Edit - CHANGE ALL command example

Detailed Command Description

The complete formats for FIND and CHANGE, showing all optional parameters, are as follows:

```
FIND  string-1  [NEXT/ALL/FIRST/LAST/PREV]
                [CHARS/PREFIX/SUFFIX/WORD]
                [X/NX]
                [col-1 [col-2]]

Command abbreviation:  F
Keyword abbreviations: PRE (for PREFIX), SUF (for SUFFIX)
```

```
CHANGE  string-1  string-2  [NEXT/ALL/FIRST/LAST/PREV]
                            [CHARS/PREFIX/SUFFIX/WORD]
                            [X/NX]
                            [col-1 [col-2]]

Command abbreviations: CHG, C
Keyword abbreviations: PRE (for PREFIX), SUF (for SUFFIX)
```

Note that FIND under edit has the same format as FIND under browse, except for the optional X/NX parameter.

The operands may be separated with blanks or commas, and may be typed in any order except that string-2 (for CHANGE) must follow string-1, and col-2 (if typed) must follow col-1. The string-1 operand is required (and string-2 for CHANGE); the others are optional.

The "string-1" operand specifies the string to be found. It may be specified in any one of the following forms:

- Any string of characters not starting or ending with an apostrophe or quotation mark, and not containing any imbedded blanks or commas.

- A delimited string: Any string starting and ending with an apostrophe (') but not containing imbedded apostrophes, or starting and ending with a quotation mark (") but not containing imbedded quotation marks.

- A hex string: Any delimited string of valid hexadecimal characters, preceded or followed with the character X. Example: X'C27B'

- A text string: Any delimited string of characters, preceded or followed with the character T. See discussion under "Use of Text Strings." Example: T'conditions for'

- A picture string: Any delimited string of picture characters, preceded or followed with the character P. See discussion under "Use of Picture Strings." Example: P'.'

- A single asterisk (*). This causes the previous value that was used as string-1 in either a FIND or CHANGE command to be used again.

The "string-2" operand is required for the CHANGE command to specify the new value of the string. The rules for coding string-2 are the same as for string-1, except as follows:

1. A text or picture string may not be specified for string-2.

2. If a single asterisk is typed, the previous value of string-2 that was used in a CHANGE command is used again.

The previous value of a character string (referenced via * or used by the Find/Change PF keys) is retained until the edit option is terminated (i.e., until return to the primary option menu).

Starting Point and Direction of Search

The starting point, direction and extent of the search may be controlled by one of the following operands.

NEXT Scan starts at the current cursor location and proceeds forwards to find the next occurrence of string-1.

ALL Scan starts at the top of data and proceeds forwards to find all occurrences of string-1.

FIRST Scan starts at the top of data and proceeds forwards to find the first occurrence of string-1.

LAST Scan starts at the bottom of data and proceeds backwards to find the last occurrence of string-1.

PREV Scan starts at the current cursor location and proceeds backwards to find the previous occurrence of string-1.

If this operand is omitted, the default is NEXT.

If the direction of the search is forward (i.e, if FIRST, ALL, or NEXT was specified), pressing the Find or Change PF key finds or changes the next occurrence of the designated string. If the direction of the search is backward (if LAST or PREV was specified), pressing these PF keys finds or changes the previous occurrence of the string. The other optional parameters (CHARS, WORD, PREFIX, SUFFIX, X, NX, and col-1, col-2) remain in effect, as specified in the last FIND or CHANGE command.

The search proceeds until one or all occurrences of string-1 are found, or until the end of data is encountered. If string-1 is not found, one of the following actions takes place.

- For FIRST, LAST, or ALL - a "NO xxxxx FOUND" message is displayed in the upper right-hand corner of the screen.
- For NEXT - a "BOTTOM OF DATA REACHED" message is displayed.
- For PREV - a "TOP OF DATA REACHED" message is displayed.

When "BOTTOM OF DATA REACHED" or "TOP OF DATA REACHED" is displayed, the user may press the Find PF key (for either FIND or CHANGE) or the Change PF key (for CHANGE) to continue searching by wrapping to the top (or bottom) of the data. If no occurrence is found anywhere in the data, a "NO xxxxx FOUND" message is displayed.

Conditions for Character String Match

The conditions for a successful "match" with string-1 may be controlled based on whether the data string begins and/or ends with a non-alphameric character (i.e., a special character or blank). The operands are: CHARS, PREFIX (may be abbreviated PRE), SUFFIX (may be abbreviated SUF), and WORD. In the following illustration, the underscored strings would be found, and the non-underscored strings would be ignored (skipped over).

CHARS 'DO'	-	DO	DONT	ADO	ADOPT	'DO'	+ADO	(DONT)	ADO-
PREFIX 'DO'	-	DO	DONT	ADO	ADOPT	'DO'	+ADO	(DONT)	ADO-
SUFFIX 'DO'	-	DO	DONT	ADO	ADOPT	'DO'	+ADO	(DONT)	ADO-
WORD 'DO'	-	DO	DONT	ADO	ADOPT	'DO'	+ADO	(DONT)	ADO-

If this operand is omitted, the default is CHARS.

Range and Column Limitations

Under edit, the lines to be searched may be limited by first using the "X" or "XX" line commands (see "Excluded Lines"), and then specifying one of the following operands on the FIND or CHANGE command:

X - Scan only lines which are excluded from display.

NX - Scan only lines which are not excluded from display.

If this operand is omitted, both excluded and non-excluded lines will be searched. When an excluded line is searched and string-1 is found, the line is automatically redisplayed (i.e., it is "popped" out from the block of excluded lines).

Note: The X/NX operand is not defined when using FIND under browse.

If the "col-1" and "col-2" operands are typed (two integers separated by a comma or at least one blank), they indicate the starting and ending column positions for the search. If col-1 is specified without col-2, the string will be found only if it starts in the specified column. If neither col-1 nor col-2 is specified, the search will continue across all columns within the current boundary columns. See description of BOUNDS commands.

Use of Text Strings

A text string, which may be used as string-1 in a FIND or CHANGE command, allows the search to be satisfied without regard to upper/lower case alphabetics. Example: FIND T'CONDITION NO. 1' would successfully find any of the following:

```
CONDITION NO. 1
Condition No. 1
condition no. 1
coNDitION nO. 1
```

An ordinary delimited string, such as: FIND 'CONDITION NO. 1' would find only the first example listed above.

Within the text string itself, is is immaterial whether alphabetics are typed in upper or lower case (even if CAPS mode is off). For example, all the following have the same effect:

```
FIND T'Edit Commands'
FIND T'EDIT COMMANDS'
FIND T'edit commands'
```

Use of Picture Strings

A picture string, which may be used as string-1 in a FIND or CHANGE command, allows the search to be satisfied when a particular type of character is encountered. Special characters are used within the picture string to represent the type of character to be found, as follows:

STRING	MEANING
P'='	- any character (don't care)
P'¬'	- any non-blank character
P'.'	- any non-displayable (invalid) character
P'#'	- any numeric character (0-9)
P'-'	- any non-numeric character
P'@'	- any alphabetic character (upper or lower case)
P'<'	- any lower case alphabetic character
P'>'	- any upper case alphabetic character
P'$'	- any special character (not alpha or numeric)

Any special characters other than the ones listed above are invalid within a picture string. A picture string may also include any alphabetic or numeric characters, in which case the characters represent themselves.

Examples of picture strings:

P'###'	- a string of three numeric characters
P'¬ ¬'	- any two non-blank characters surrounding a blank
P'.'	- any non-displayable character
P' #'	- a blank followed by a numeric character
P'#AB'	- a numeric character followed by 'AB'

Examples of FIND and CHANGE commands using picture strings:

FIND P'.'	- find next non-displayable character
FIND P'¬' 72	- find next non-blank character in column 72
C ALL P'=' ' ' 73 80	- change any characters in columns 73-80 to blanks
F P' ¬' 1	- find the next line with a blank in column 1 followed by a non-blank

When the picture string P'.' is used to find or change a non-displayable character, the hexadecimal representation of the character is shown in the confirmation message which appears in the upper right-hand corner of the screen.

ADVANCED FEATURES

The following commands are described in this section.

Line commands:

MASK
< (data left)
> (data right)
((columns left)
) (columns right)
BOUNDS
X (exclude)
S (show)
F (first)
L (last)
TABS

Primary commands:

COPY
MOVE
CREATE
REPLACE
NULLS
TABS
PROFILE
HEX

Mask Definition

The mask is used in conjunction with the "I" (insert), "TE" (text entry), and "TS" (text split) line commands. Whenever a new line is inserted, it is pre-filled with the contents of the mask. Initially, the mask contains all blanks.

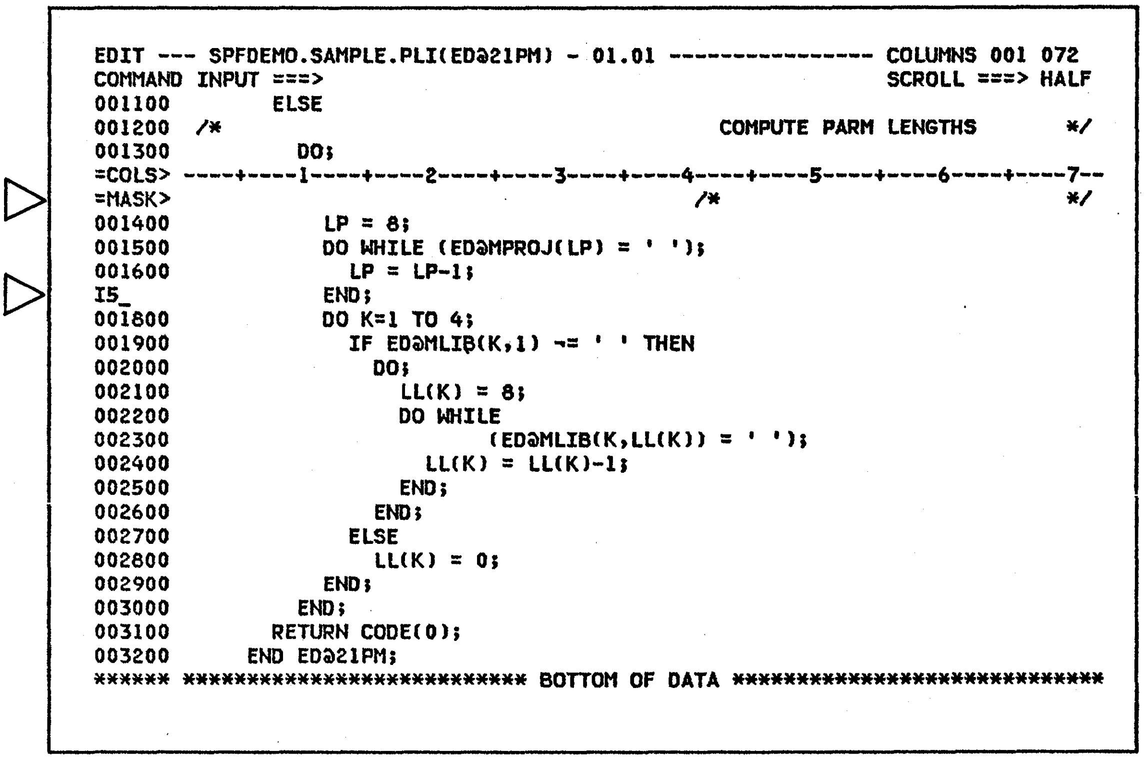
To display the mask, the characters "MASK" are entered in the line command area, overtyping the line number. When the ENTER key is pressed, the mask is displayed at the designated position.

```
MASK  - Display mask definition line
```

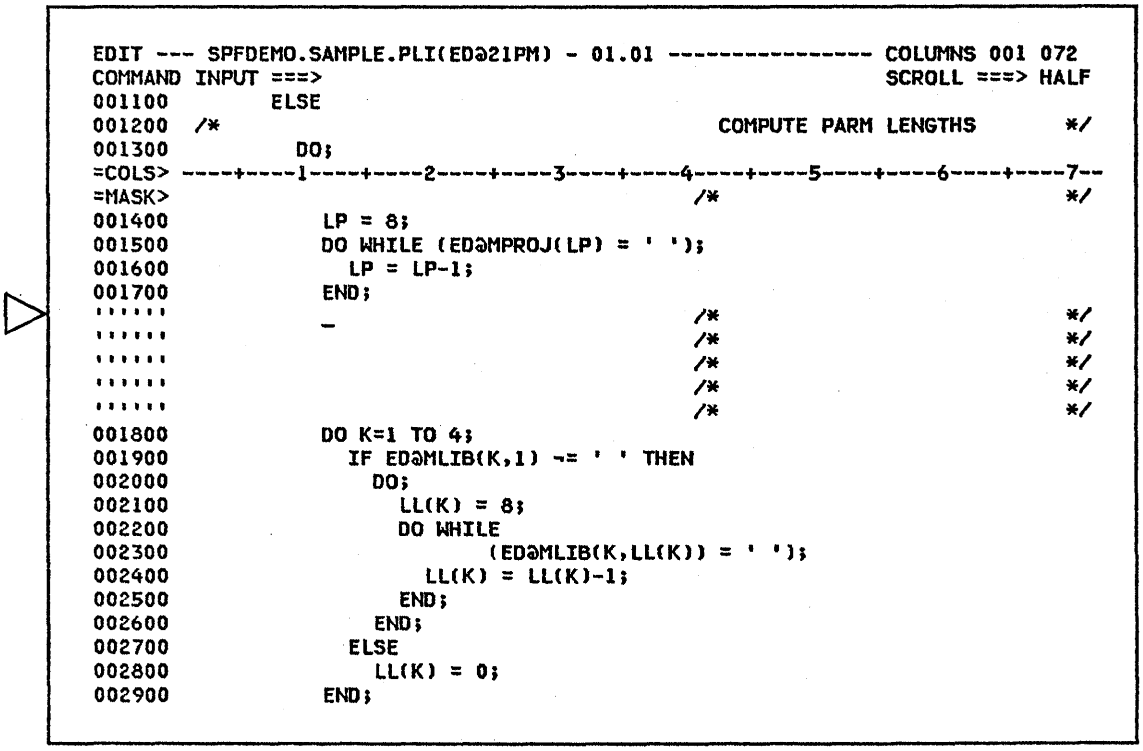
Once displayed, the mask may be changed to allow pre-entry of data on inserted lines. An example is shown in Figure 32. In this example, the mask has been displayed and the characters "/*" and "*/" have been entered at the desired positions on the mask line. Five new lines are then inserted following line 1700. The second part of the example shows that the new lines contain the contents of the mask.

The mask line may be removed from display by entering a "D" in the line command area, or by entering a RESET primary command. The mask line is never saved as part of the data.

The mask remains in effect (even if it is not displayed) until it is again changed by the user. The contents of the mask are retained in the current edit profile, and automatically used the next time the user edits the same type of data.



```
EDIT --- SPFDEMO.SAMPLE.PLI(ED@21PM) - 01.01 ---------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
001100        ELSE
001200  /*                                       COMPUTE PARM LENGTHS        */
001300          DO;
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
=MASK>                                         /*                            */
001400            LP = 8;
001500            DO WHILE (ED@MPROJ(LP) = ' ');
001600              LP = LP-1;
I5_               END;
001800            DO K=1 TO 4;
001900              IF ED@MLIB(K,1) ¬= ' ' THEN
002000                DO;
002100                  LL(K) = 8;
002200                  DO WHILE
002300                         (ED@MLIB(K,LL(K)) = ' ');
002400                    LL(K) = LL(K)-1;
002500                  END;
002600                END;
002700              ELSE
002800                LL(K) = 0;
002900            END;
003000          END;
003100        RETURN CODE(0);
003200      END ED@21PM;
****** *************************** BOTTOM OF DATA ****************************
```



```
EDIT --- SPFDEMO.SAMPLE.PLI(ED@21PM) - 01.01 ---------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
001100        ELSE
001200  /*                                       COMPUTE PARM LENGTHS        */
001300          DO;
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
=MASK>                                         /*                            */
001400            LP = 8;
001500            DO WHILE (ED@MPROJ(LP) = ' ');
001600              LP = LP-1;
001700            END;
''''''                                         /*                            */
''''''            _                            /*                            */
''''''                                         /*                            */
''''''                                         /*                            */
''''''                                         /*                            */
001800            DO K=1 TO 4;
001900              IF ED@MLIB(K,1) ¬= ' ' THEN
002000                DO;
002100                  LL(K) = 8;
002200                  DO WHILE
002300                         (ED@MLIB(K,LL(K)) = ' ');
002400                    LL(K) = LL(K)-1;
002500                  END;
002600                END;
002700              ELSE
002800                LL(K) = 0;
002900            END;
```

Figure 32. Edit - define mask example

Indentation Changes

The contents of a line (or block of lines) may be shifted left or right by means of the shift line commands. The two forms of these commands are:

">" and "<"	for "data" shifting
")" and "("	for "column" shifting

Data shifting attempts to shift the body of a program statement without shifting the label or comments, and will prevent loss of data. Column shifting shifts information without regard to its contents, and may result in the loss of data. (See further discussion below.)

To shift the contents of a line left, the single character "<" or "(" is entered in the line command area, overtyping the line number. When the ENTER key is pressed, the data in the line containing the command is shifted left by two column positions (two is the default for shift operations).

<	- Data shift line left 2 positions
<n	- Data shift line left "n" positions
<<	- Data shift block of lines left 2 positions
<<n	- Data shift block of lines left "n" positions

(	- Col. shift line left 2 positions
(n	- Col. shift line left "n" positions
((	- Col. shift block of lines left 2 positions
((n	- Col. shift block of lines left "n" positions

To shift a block of lines left, "<<" or "((" is entered in the line command area of the first and last lines to be shifted. The first and last lines need not be on the same page; scrolling may be used between entering the first command and the second command.

The equivalent commands for right shifting are ">", ")", ">>" and "))".

>	- Data shift line right 2 positions
>n	- Data shift line right "n" positions
>>	- Data shift block of lines right 2 positions
>>n	- Data shift block of lines right "n" positions

)	- Col. shift line right 2 positions
)n	- Col. shift line right "n" positions
))	- Col. shift block of lines right 2 positions
))n	- Col. shift block of lines right "n" positions

A number may be entered following any of the shift commands to indicate the number of column positions to be shifted. For example, ">>5" entered on the first or last (or both) lines to be shifted would cause a right shift of five column positions.

Shifting occurs within column boundaries that may be changed by the user. Default column boundaries are established by edit, based on the data "type" (SPF library type or CMS filetype). The default boundaries are normally the first and last columns where source text may be entered for the particular programming language. See BOUNDS command for a discussion of default boundaries and the procedures for changing them.

Data Shifting - ">" and "<"

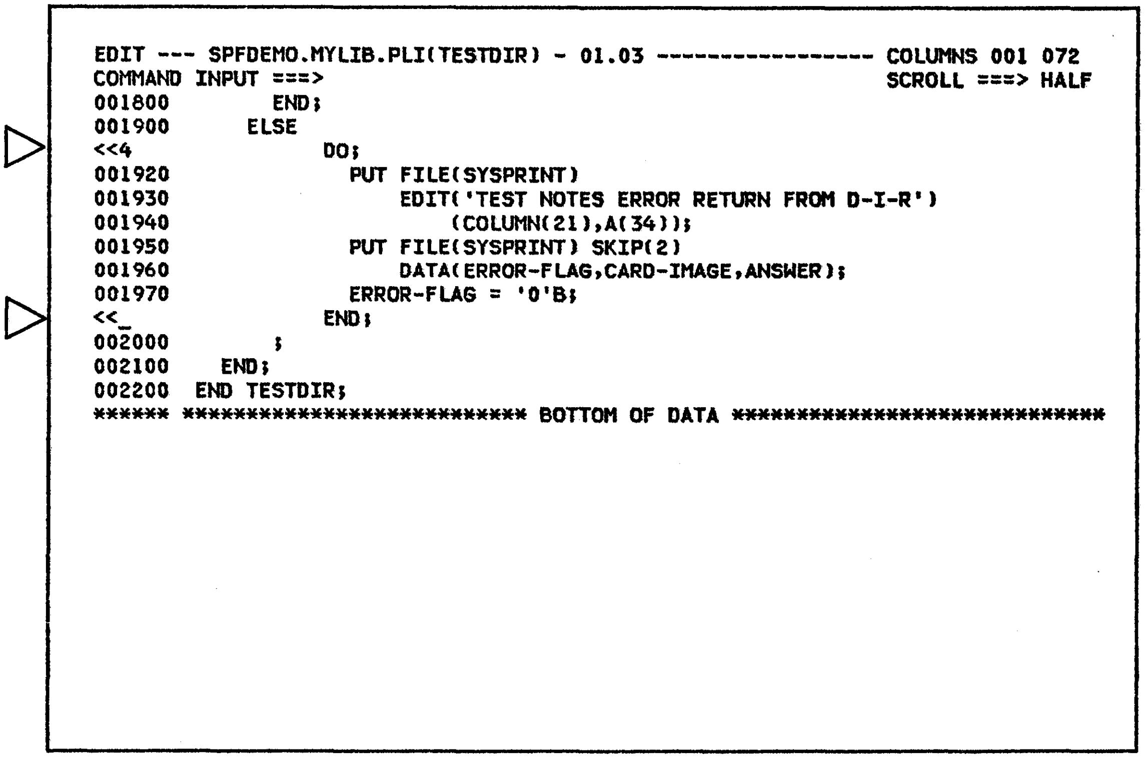
Within the column boundaries, data shifting is accomplished by squeezing out multiple blank characters at one end of the data, and increasing the number of blanks at the other end. The following rules are followed in performing a data shift.

- Non-blank characters are never deleted or truncated.
- A single blank is never deleted.
- Within apostrophes, the number of multiple blanks is never changed.
- Non-blank characters are never shifted into or out of the left column boundary position (normally the label field).
- If a shift cannot be completed, it is partially performed and the line number is replaced with the following intensified warning message: ==ERR>. The warning message may be removed via the RESET primary command, or by overtyping the message or data on that line.

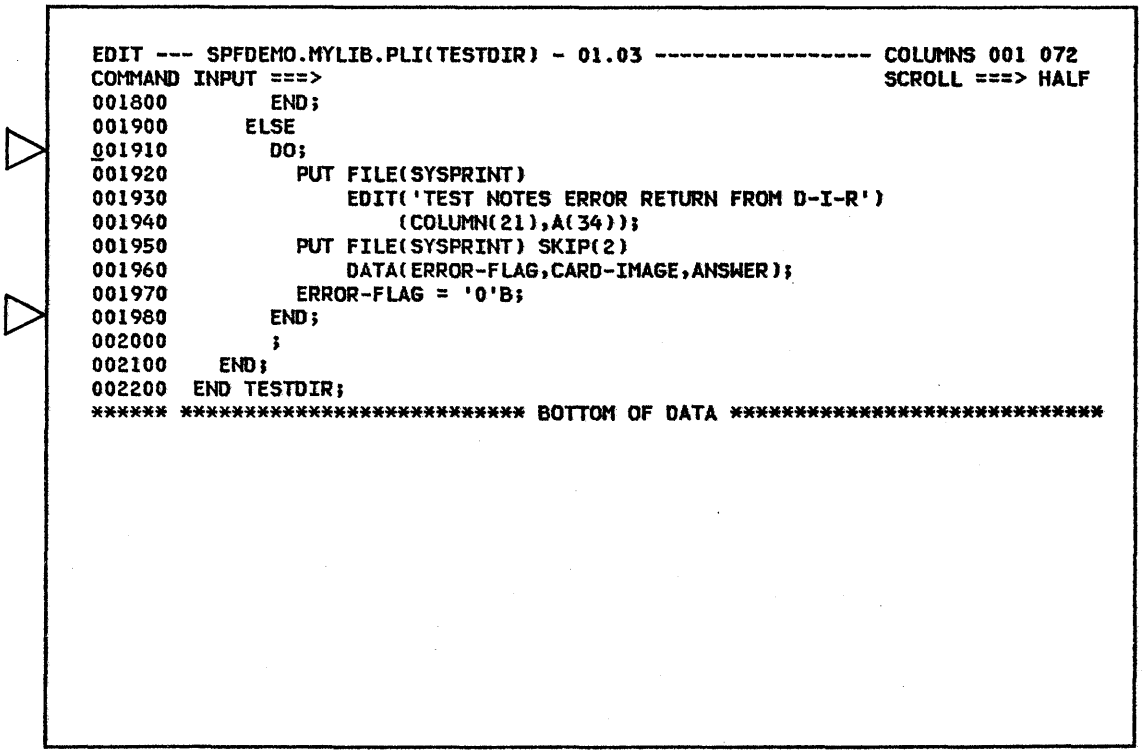
Figure 33 shows a before-and-after example of data shifting. The block of lines, starting at line 1910 through line 1980, is shifted left four column positions.

Column Shifting - ")" and "("

Column shifting is accomplished by simply moving all of the characters (within the column boundaries) left or right by the designated number of positions. There is no squeezing or increasing of blanks. There is no restriction on shifting non-blank characters into or out of a column boundary position. Characters shifted beyond a column boundary postion are lost, with no warning message.



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                             SCROLL ===> HALF
001800        END;
001900      ELSE
<<4               DO;
001920              PUT FILE(SYSPRINT)
001930                  EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001940                      (COLUMN(21),A(34));
001950              PUT FILE(SYSPRINT) SKIP(2)
001960                  DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001970              ERROR-FLAG = '0'B;
<<_               END;
002000        ;
002100    END;
002200  END TESTDIR;
****** *************************** BOTTOM OF DATA ******************************
```



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                             SCROLL ===> HALF
001800        END;
001900      ELSE
001910        DO;
001920          PUT FILE(SYSPRINT)
001930              EDIT('TEST NOTES ERROR RETURN FROM D-I-R')
001940                  (COLUMN(21),A(34));
001950          PUT FILE(SYSPRINT) SKIP(2)
001960              DATA(ERROR-FLAG,CARD-IMAGE,ANSWER);
001970          ERROR-FLAG = '0'B;
001980        END;
002000        ;
002100    END;
002200  END TESTDIR;
****** *************************** BOTTOM OF DATA ******************************
```

Figure 33. Edit - left shift data example

Boundary Column Definition

To display the column boundary definition line, the characters "BOUNDS" (or an acceptable abbreviation) are entered in the line command area, overtyping the line number. When the ENTER key is pressed, the boundary definition line is inserted at the designated position.

BOUNDS - Display boundary definition line
Command abbreviations: BOUND, BNDS, BND

The column boundaries are used to limit the scope of:

- Left and right shift line commands
- FIND and CHANGE commands when explicit columns were not specified by the user
- Overlay line command
- Text entry, text split, and text flow line commands
- Left and right scrolling.

The effect on left and right scrolling is as follows:

1. A left scroll will stop at the left bound, and a right scroll will stop at the right bound. A subsequent left or right scroll will go beyond the bound (assuming the bound is not at the left-most or right-most column position).

2. The initial display (when edit is entered) will be pre-scrolled so that the left boundary column is the first column displayed.

The boundaries are initialized by edit based on the data "type" (SPF library type or CMS filetype) and whether or not the data is sequence numbered. The default left and right column boundaries are shown in Figure 34.

RECORD FORMAT	DATA TYPE	NUMBERED	UNNUMBERED
FIXED LENGTH	ASM or ASSEMBLE	1, 71	1, 71
	COBOL	7, 72	7, 72
	(OTHER)	1, LRECL-8	1, LRECL
VARIABLE LENGTH	(ANY)	9, LRECL	1, LRECL

Figure 34. Edit - default column boundary settings

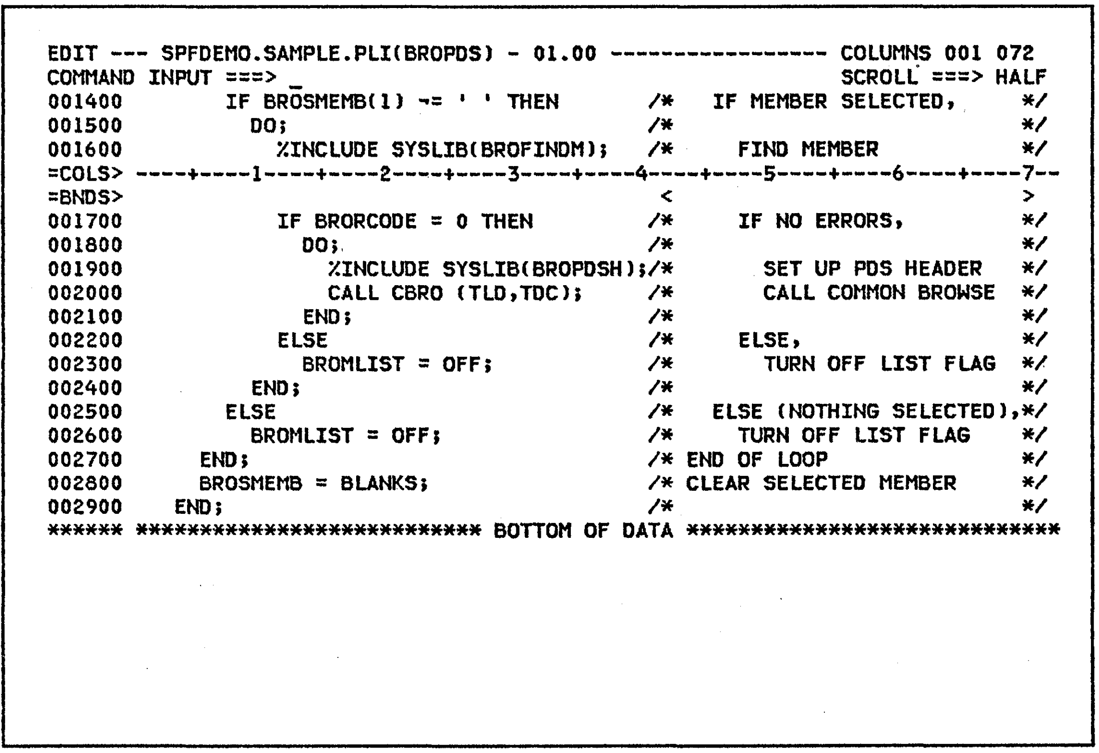
The user may change the boundaries by displaying the boundary definition line and changing its contents. The boundary definition line contains a single "<" character indicating the position of the left column boundary, and a single ">" character indicating the position of the right column boundary. If the user has already changed the boundaries, he may revert to the default bounds by displaying the boundary definition line and blanking out its contents. The ERASE EOF key may be used for this purpose.

If the default boundaries are in effect, they are automatically adjusted whenever NUMBER mode is turned on or off. If the user has changed the bounds from the default settings, they are not affected by the setting of NUMBER mode.

Figure 35 shows an example of the boundary definition line displayed together with the column identification line. In this example, the left bound is currently set at column 42 and the right bound at column 70.

The boundary definition line may be removed from display by entering a "D" in the line command area, or by entering a RESET primary command. The boundary definition line is never saved as part of the data.

The boundary definition line is retained in the current edit profile, and automatically used the next time the user edits the same type of data.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.00 ----------------- COLUMNS 001 072
COMMAND INPUT ===> _                                          SCROLL ===> HALF
001400        IF BROSMEMB(1) ¬= ' ' THEN       /*   IF MEMBER SELECTED,     */
001500          DO;                            /*                           */
001600            %INCLUDE SYSLIB(BROFINDM);   /*     FIND MEMBER           */
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
=BNDS>                                          <                           >
001700            IF BRORCODE = 0 THEN         /*     IF NO ERRORS,         */
001800              DO;                        /*                           */
001900                %INCLUDE SYSLIB(BROPDSH);/*       SET UP PDS HEADER   */
002000                CALL CBRO (TLD,TDC);     /*       CALL COMMON BROWSE  */
002100              END;                       /*                           */
002200            ELSE                         /*     ELSE,                 */
002300              BROMLIST = OFF;            /*       TURN OFF LIST FLAG  */
002400          END;                           /*                           */
002500        ELSE                             /*   ELSE (NOTHING SELECTED),*/
002600          BROMLIST = OFF;                /*     TURN OFF LIST FLAG    */
002700      END;                               /* END OF LOOP               */
002800      BROSMEMB = BLANKS;                 /* CLEAR SELECTED MEMBER     */
002900    END;                                 /*                           */
****** *************************** BOTTOM OF DATA *****************************
```

Figure 35. Edit - boundary definition line

Excluded Lines

When editing a program segment which exceeds the screen data size, it is often difficult to determine whether the control structure and indentation levels are correct. The excluded lines capability allows blocks of lines to be collapsed so that the overall control structure may be viewed. The lines are excluded from display but are not deleted in the data.

Excluded lines may also be used in conjunction with the "X" and "NX" options of the FIND and CHANGE commands to limit the scope of search, and/or to easily identify lines which have been found or changed by causing them to "pop out" of a block of excluded lines.

To exclude a line, the single character "X" is entered in the line command area, overtyping the line number. A number may follow the "X" to indicate that more than one line is to be excluded. For example, "X5" would cause five lines to be excluded, starting with the line containing the "X5". When the ENTER key is pressed, the line(s) are replaced with a single message line, which indicates how many lines were excluded.

X	- Exclude line
Xn	- Exclude "n" lines
XX	- Exclude block of lines

To exclude a block of lines, the double character "XX" is entered in the line command area of the first and last lines to be excluded. The first and last lines need not be on the same page.

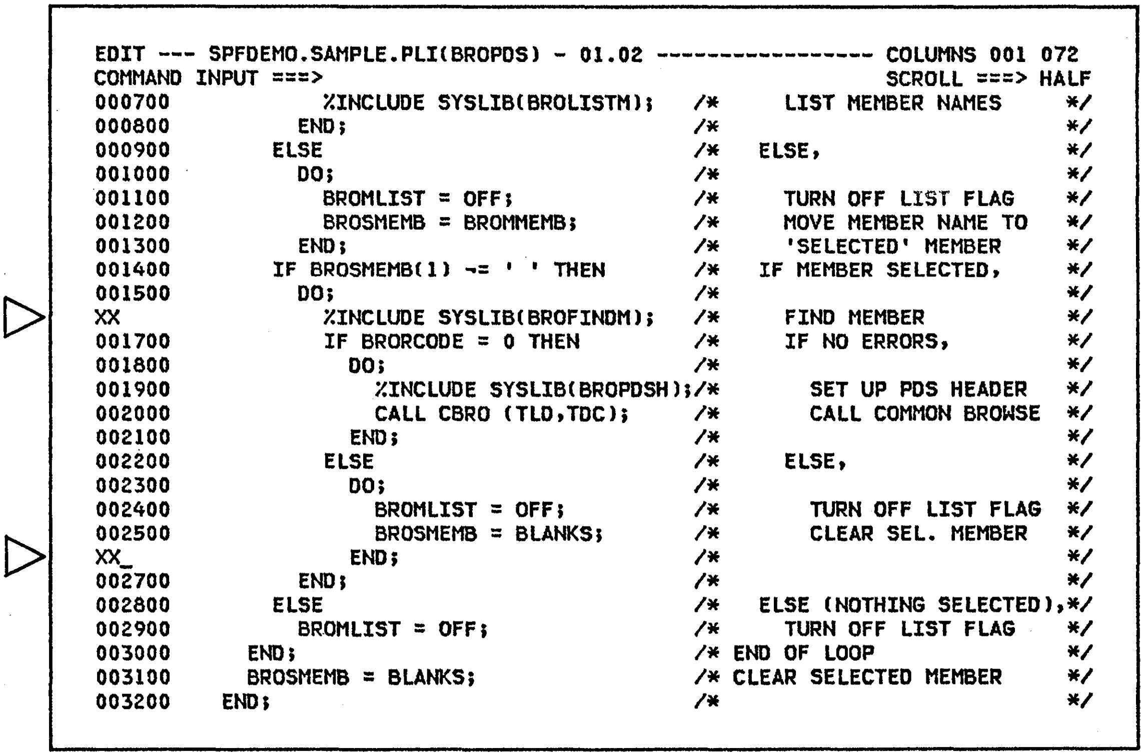
Figure 36 shows a before-and-after example of excluded lines. The block of lines, starting at line 1600 through line 2600, is excluded from display and replaced with a single message line.

Excluded lines may be redisplayed by entering the RESET primary command. Alternatively, one or more excluded lines may be redisplayed by entering "S" (show), "F" (first), or "L" (last) in the line command area of the "EXCLUDED LINES" message, overtyping the dashes which are normally in that area. Each of these line commands may be followed by a number to cause redisplay of more than one line.

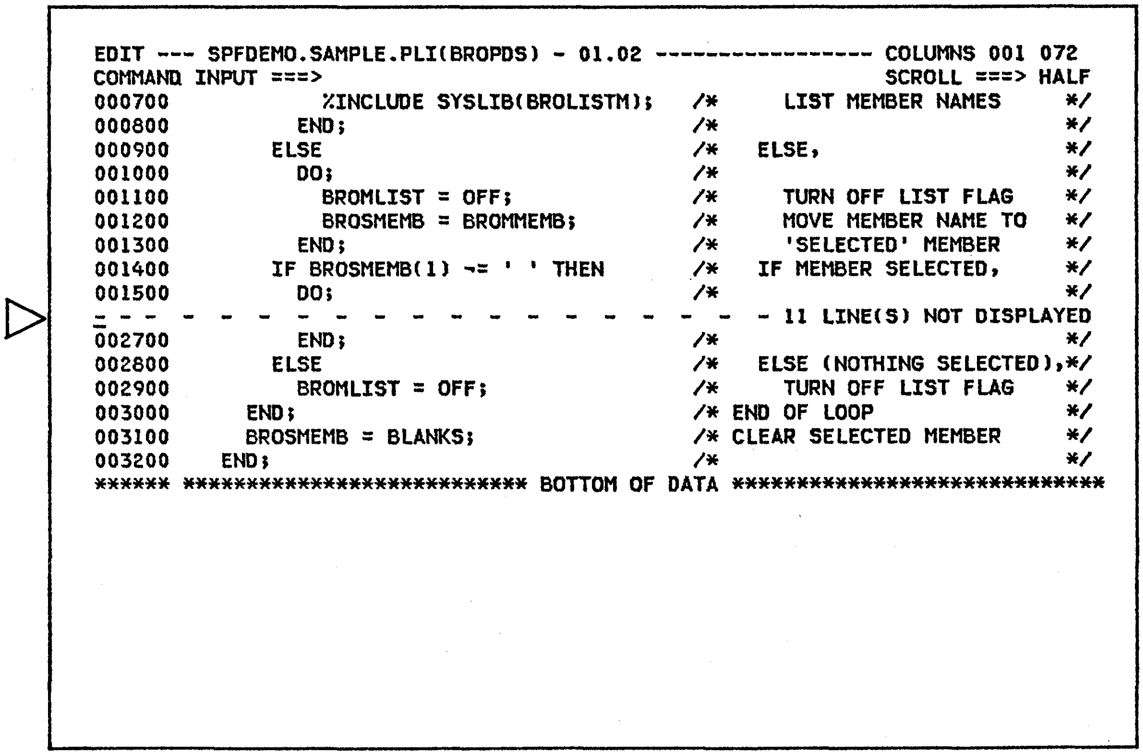
S	- Show line
Sn	- Show "n" lines

F	- Show first line
Fn	- Show first "n" lines

L	- Show last line
Ln	- Show last "n" lines



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000700            %INCLUDE SYSLIB(BROLISTM);    /*     LIST MEMBER NAMES     */
000800          END;                            /*                           */
000900        ELSE                              /*   ELSE,                   */
001000          DO;                             /*                           */
001100            BROMLIST = OFF;               /*     TURN OFF LIST FLAG    */
001200            BROSMEMB = BROMMEMB;          /*     MOVE MEMBER NAME TO   */
001300          END;                            /*     'SELECTED' MEMBER     */
001400        IF BROSMEMB(1) ¬= ' ' THEN        /*   IF MEMBER SELECTED,     */
001500          DO;                             /*                           */
XX                %INCLUDE SYSLIB(BROFINDM);    /*     FIND MEMBER           */
001700            IF BRORCODE = 0 THEN          /*     IF NO ERRORS,         */
001800              DO;                         /*                           */
001900                %INCLUDE SYSLIB(BROPDSH);/*       SET UP PDS HEADER   */
002000                CALL CBRO (TLD,TDC);      /*       CALL COMMON BROWSE  */
002100              END;                        /*                           */
002200            ELSE                          /*     ELSE,                 */
002300              DO;                         /*                           */
002400                BROMLIST = OFF;           /*       TURN OFF LIST FLAG  */
002500                BROSMEMB = BLANKS;        /*       CLEAR SEL. MEMBER   */
XX_                 END;                        /*                           */
002700          END;                            /*                           */
002800        ELSE                              /*   ELSE (NOTHING SELECTED),*/
002900          BROMLIST = OFF;                 /*     TURN OFF LIST FLAG    */
003000      END;                                /* END OF LOOP               */
003100      BROSMEMB = BLANKS;                  /* CLEAR SELECTED MEMBER     */
003200    END;                                  /*                           */
```



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000700            %INCLUDE SYSLIB(BROLISTM);    /*     LIST MEMBER NAMES     */
000800          END;                            /*                           */
000900        ELSE                              /*   ELSE,                   */
001000          DO;                             /*                           */
001100            BROMLIST = OFF;               /*     TURN OFF LIST FLAG    */
001200            BROSMEMB = BROMMEMB;          /*     MOVE MEMBER NAME TO   */
001300          END;                            /*     'SELECTED' MEMBER     */
001400        IF BROSMEMB(1) ¬= ' ' THEN        /*   IF MEMBER SELECTED,     */
001500          DO;                             /*                           */
- - -  - - - - - - - - - - - - - - - - - - - - - 11 LINE(S) NOT DISPLAYED
002700          END;                            /*                           */
002800        ELSE                              /*   ELSE (NOTHING SELECTED),*/
002900          BROMLIST = OFF;                 /*     TURN OFF LIST FLAG    */
003000      END;                                /* END OF LOOP               */
003100      BROSMEMB = BLANKS;                  /* CLEAR SELECTED MEMBER     */
003200    END;                                  /*                           */
****** *************************** BOTTOM OF DATA ****************************
```

Figure 36. Edit - excluded lines example

The "S" line command causes the block of excluded lines to be scanned, and one or more lines selected for redisplay. The selected lines are those with the left-most indentation levels (i.e, with the fewest number of leading blanks). If "S3" is entered, for example, the three lines with the left-most indentation level will be redisplayed. If more than three lines exist at this indentation level, the first three will be displayed.

The "F" line command causes one or more lines at the beginning of the block to be redisplayed. The "L" line command causes one or more lines at the end of the block to be redisplayed.

Figure 37 shows a before-and-after example of redisplaying excluded lines. The command "S3" is entered in the line command area of the "EXCLUDED LINES" message. Three lines with the highest indentation level are redisplayed.

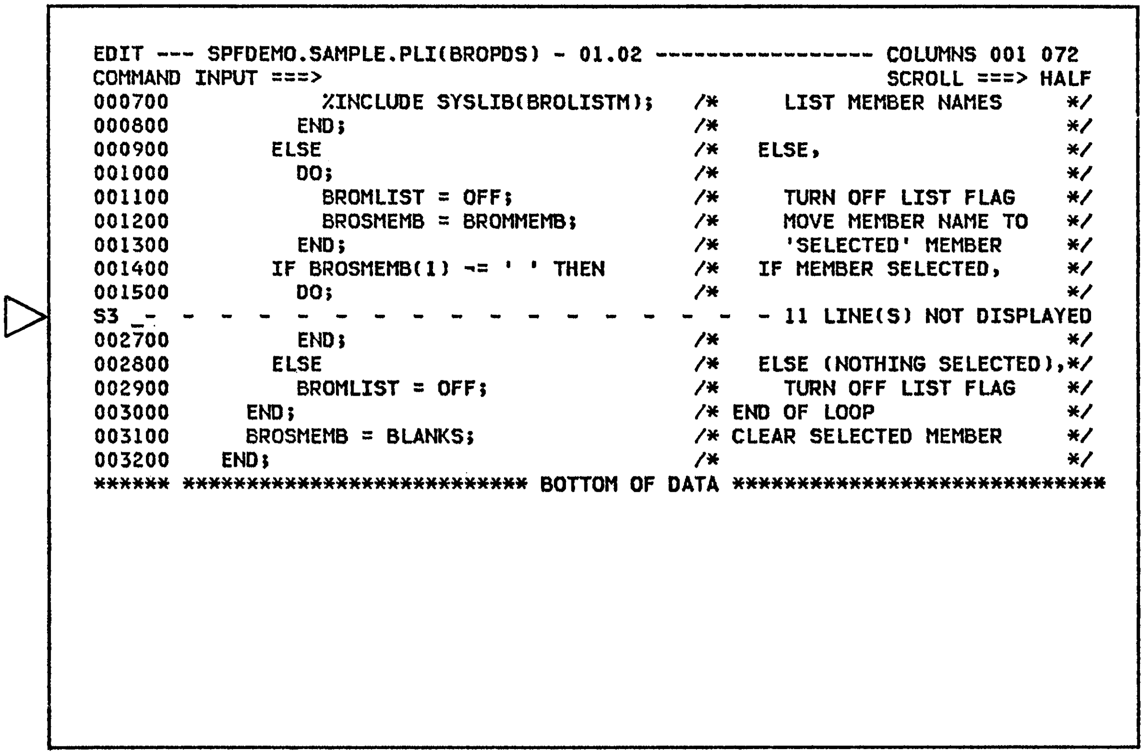
Any line command that normally operates on a single line may be entered in the line command area of the "EXCLUDED LINES" message. For example, a "D" entered on that line will delete the entire block of excluded lines. This feature is particularly useful in conjunction with the shift commands. Suppose, for example, you want to "data" shift all lines to the left by 4 column positions. Proceed as follows:

1. On the first data line, enter "X99999". This will exclude all lines.

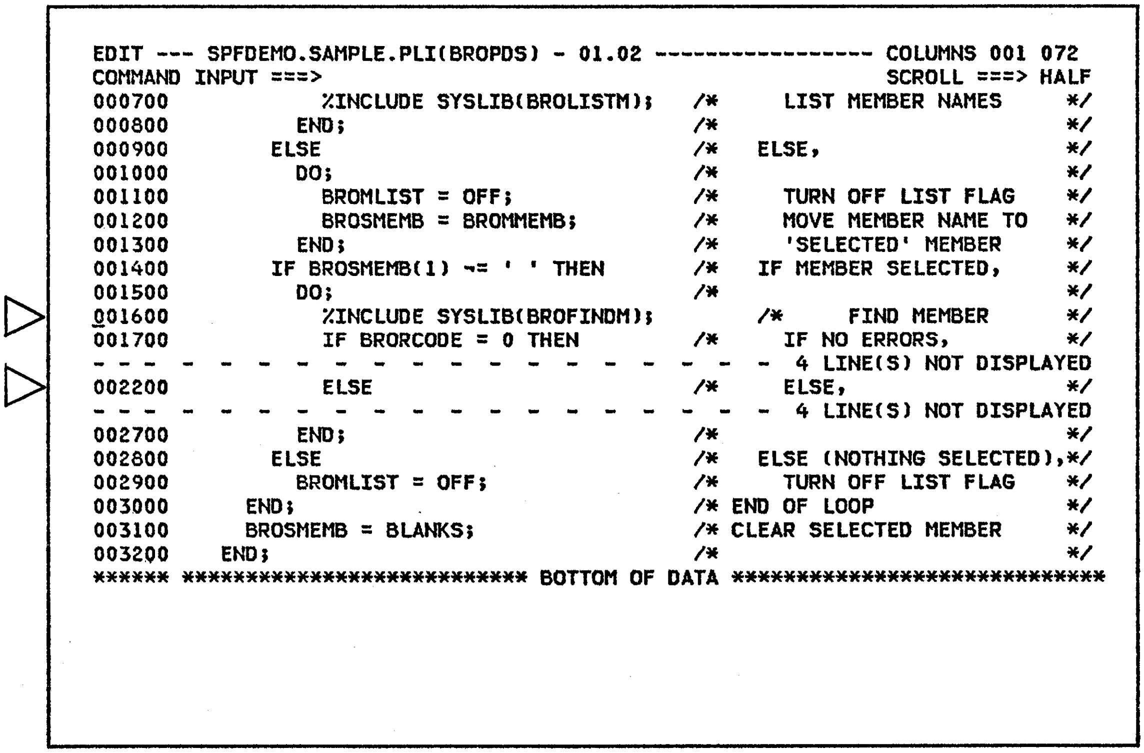
2. On the "EXCLUDED LINES" message, enter the shift command "<4".

3. In the primary command area, enter the RESET command to bring the lines back into view.

Note: Excluded lines need not be redisplayed before saving the data. The "EXCLUDED LINES" message line is never saved as part of the data.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000700            %INCLUDE SYSLIB(BROLISTM);   /*    LIST MEMBER NAMES     */
000800          END;                           /*                          */
000900        ELSE                             /*  ELSE,                   */
001000          DO;                            /*                          */
001100            BROMLIST = OFF;              /*    TURN OFF LIST FLAG    */
001200            BROSMEMB = BROMMEMB;         /*    MOVE MEMBER NAME TO   */
001300          END;                           /*    'SELECTED' MEMBER     */
001400        IF BROSMEMB(1) ¬= ' ' THEN       /*  IF MEMBER SELECTED,     */
001500          DO;                            /*                          */
S3 _- - - - - - - - - - - - - - - - - - - - - - - - 11 LINE(S) NOT DISPLAYED
002700          END;                           /*                          */
002800        ELSE                             /*  ELSE (NOTHING SELECTED),*/
002900          BROMLIST = OFF;                /*    TURN OFF LIST FLAG    */
003000      END;                               /* END OF LOOP              */
003100      BROSMEMB = BLANKS;                 /* CLEAR SELECTED MEMBER    */
003200    END;                                 /*                          */
****** *************************** BOTTOM OF DATA ****************************
```



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000700            %INCLUDE SYSLIB(BROLISTM);   /*    LIST MEMBER NAMES     */
000800          END;                           /*                          */
000900        ELSE                             /*  ELSE,                   */
001000          DO;                            /*                          */
001100            BROMLIST = OFF;              /*    TURN OFF LIST FLAG    */
001200            BROSMEMB = BROMMEMB;         /*    MOVE MEMBER NAME TO   */
001300          END;                           /*    'SELECTED' MEMBER     */
001400        IF BROSMEMB(1) ¬= ' ' THEN       /*  IF MEMBER SELECTED,     */
001500          DO;                            /*                          */
001600            %INCLUDE SYSLIB(BROFINDM);      /*     FIND MEMBER       */
001700            IF BRORCODE = 0 THEN         /*    IF NO ERRORS,         */
- - - - - - - - - - - - - - - - - - - - - - - - - - 4 LINE(S) NOT DISPLAYED
002200            ELSE                         /*    ELSE,                 */
- - - - - - - - - - - - - - - - - - - - - - - - - - 4 LINE(S) NOT DISPLAYED
002700          END;                           /*                          */
002800        ELSE                             /*  ELSE (NOTHING SELECTED),*/
002900          BROMLIST = OFF;                /*    TURN OFF LIST FLAG    */
003000      END;                               /* END OF LOOP              */
003100      BROSMEMB = BLANKS;                 /* CLEAR SELECTED MEMBER    */
003200    END;                                 /*                          */
****** *************************** BOTTOM OF DATA ****************************
```

Figure 37. Edit - show lines example

Data Merging and Segmentation

Merging or segmentation of data is accomplished by moving or copying lines into or out of the data being edited. To merge data, a primary command is used to specify the source of the move or copy, and a line command is used to specify the destination. To segment data, the process is reversed.

Merging Data

The COPY and MOVE primary commands (not be confused with the line commands of the same name) are used to specify a sequential file or member of an SPF library, MACLIB, or TXTLIB to be copied into the data being edited. If the MOVE command is used, the member or file is deleted (erased) following a successful read operation. (For a concatenated sequence of SPF libraries, the delete occurs only if the member was in the first library of the concatentation sequence.) These commands are of the form:

```
COPY  [member-name]
```

```
MOVE  [member-name]
```

If the "member-name" operand is entered, the member is fetched from the library (or concatenated sequence of libraries) currently being edited. If the operand is omitted, a menu will be displayed to allow specification of a file or library member as the source for the COPY or MOVE.

The destination for the COPY or MOVE is specified by the "A" (after) or "B" (before) line command in the data being edited. The "A" or "B" line command need not be entered if the data being edited contains no lines (new or member of file). Note that a number may not follow the "A" or "B" line command when it is used in conjunction with a COPY or MOVE primary command.

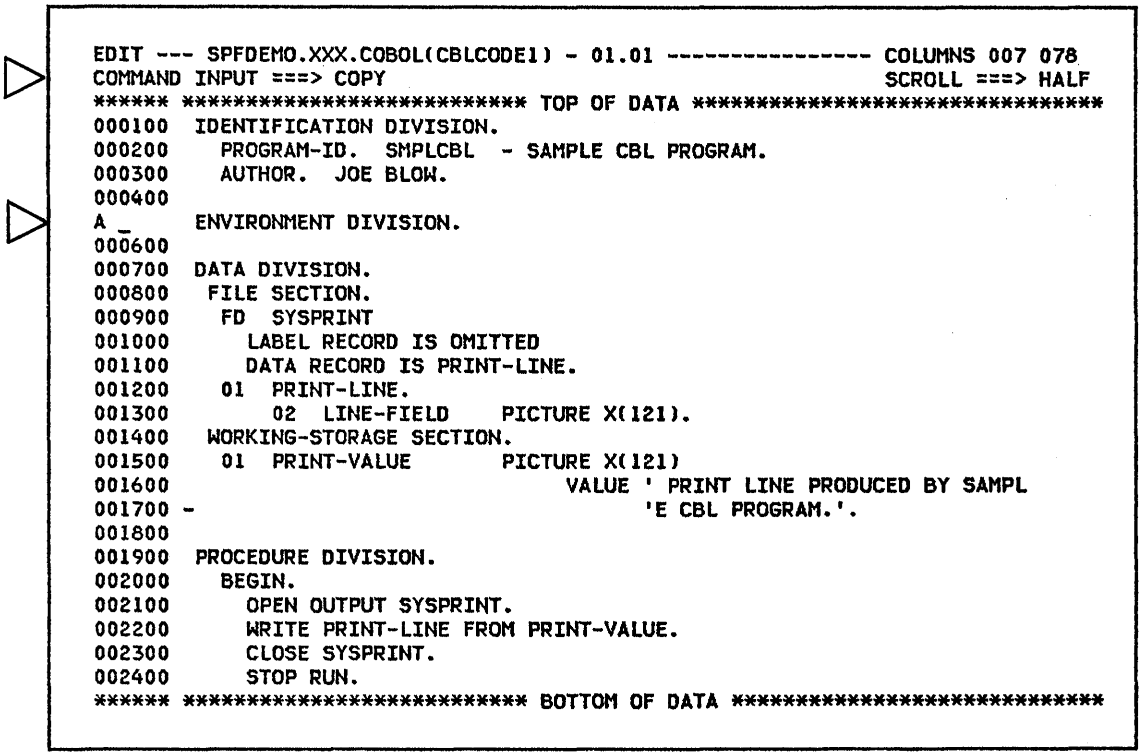
If the member-name operand is omitted on the COPY or MOVE primary command, the "A" or "B" line command must be entered before the file menu will be displayed (unless the data being edited contains no lines).

The COPY menu allows a range of lines in the source data set to be specified. The line searching criteria must also be specified, as follows:

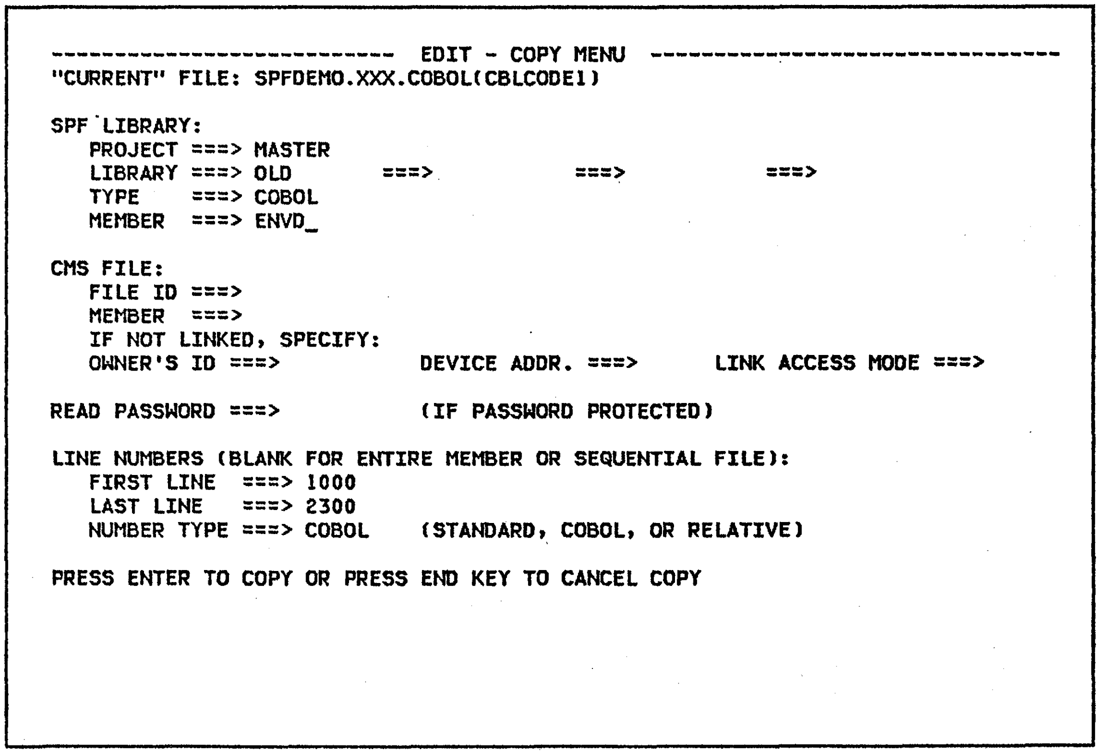
- STANDARD - search for 8-character "standard" line numbers
- COBOL - search for 6-character COBOL line numbers
- RELATIVE - ignore line numbers in the data, and interpret the specified line numbers as relative to the start of the data.

See "Sequence Number Generation and Control" for a further discussion of standard and COBOL line numbers.

Figure 38 shows an example of the COPY command. Member CBLCODE1 is being edited in library SPFDEMO.XXX.COBOL. A COPY command with no operand is entered in the primary command area, and an "A" line command is entered at line 500. When the ENTER key is pressed, the COPY menu is displayed (second half of the figure). In this example, lines 1000 through 2300 of member ENVD in library MASTER.OLD.COBOL are to be copied. When the menu has been filled in and the ENTER key pressed, the lines will be copied and inserted after line 500, and the edit data will be redisplayed.



```
EDIT --- SPFDEMO.XXX.COBOL(CBLCODE1) - 01.01 ---------------- COLUMNS 007 078
COMMAND INPUT ===> COPY                                       SCROLL ===> HALF
****** *************************** TOP OF DATA ********************************
000100  IDENTIFICATION DIVISION.
000200    PROGRAM-ID.  SMPLCBL  - SAMPLE CBL PROGRAM.
000300    AUTHOR.  JOE BLOW.
000400
A _     ENVIRONMENT DIVISION.
000600
000700  DATA DIVISION.
000800   FILE SECTION.
000900    FD  SYSPRINT
001000      LABEL RECORD IS OMITTED
001100      DATA RECORD IS PRINT-LINE.
001200    01  PRINT-LINE.
001300        02  LINE-FIELD    PICTURE X(121).
001400   WORKING-STORAGE SECTION.
001500    01  PRINT-VALUE       PICTURE X(121)
001600                               VALUE ' PRINT LINE PRODUCED BY SAMPL
001700 -                                     'E CBL PROGRAM.'.
001800
001900  PROCEDURE DIVISION.
002000    BEGIN.
002100      OPEN OUTPUT SYSPRINT.
002200      WRITE PRINT-LINE FROM PRINT-VALUE.
002300      CLOSE SYSPRINT.
002400      STOP RUN.
****** *************************** BOTTOM OF DATA *****************************
```



```
--------------------------- EDIT - COPY MENU --------------------------------
"CURRENT" FILE: SPFDEMO.XXX.COBOL(CBLCODE1)

SPF LIBRARY:
   PROJECT ===> MASTER
   LIBRARY ===> OLD       ===>           ===>           ===>
   TYPE    ===> COBOL
   MEMBER  ===> ENVD_

CMS FILE:
   FILE ID ===>
   MEMBER  ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>          (IF PASSWORD PROTECTED)

LINE NUMBERS (BLANK FOR ENTIRE MEMBER OR SEQUENTIAL FILE):
   FIRST LINE  ===> 1000
   LAST LINE   ===> 2300
   NUMBER TYPE ===> COBOL   (STANDARD, COBOL, OR RELATIVE)

PRESS ENTER TO COPY OR PRESS END KEY TO CANCEL COPY
```

Figure 38. Edit - COPY command example

The CREATE and REPLACE primary commands are used to specify a sequential file or member of an SPF library, MACLIB, or TXTLIB to be written from the data being edited. CREATE will perform a check to ensure that the specified member or file does not already exist. REPLACE will either create the member or file (if it does not already exist) or replace it (if it does exist). The commands are of the form:

CREATE [member-name]
Command abbreviation: CRE

REPLACE [member-name]
Command abbreviations: REPL, REP

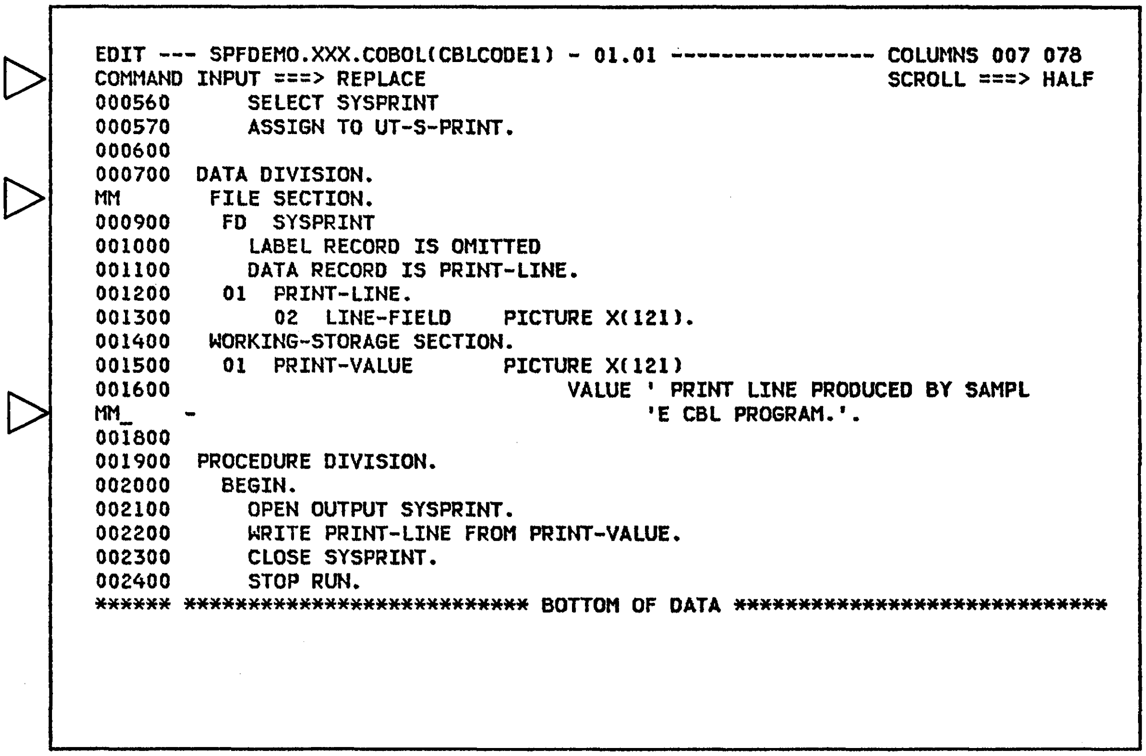
If the "member-name" operand is entered, the member will be written in the library currently being edited. (For a concatenated sequence of SPF libraries, the member is always written in the first library of the concatenation sequence.) If the operand is omitted, a menu will be displayed to allow specification of the destination of the operation.

The source of the operation is specified by the "C" (copy) or "M" (move) line commands. The block form -- "CC" or "MM" -- may be used to specify the first and last lines. If the "M" or "MM" line command is used, the lines are deleted from the data being edited following a successful write operation. The form "Cn" or "Mn", where "n" is a large number, may be used on the first data line to copy or move all the data.

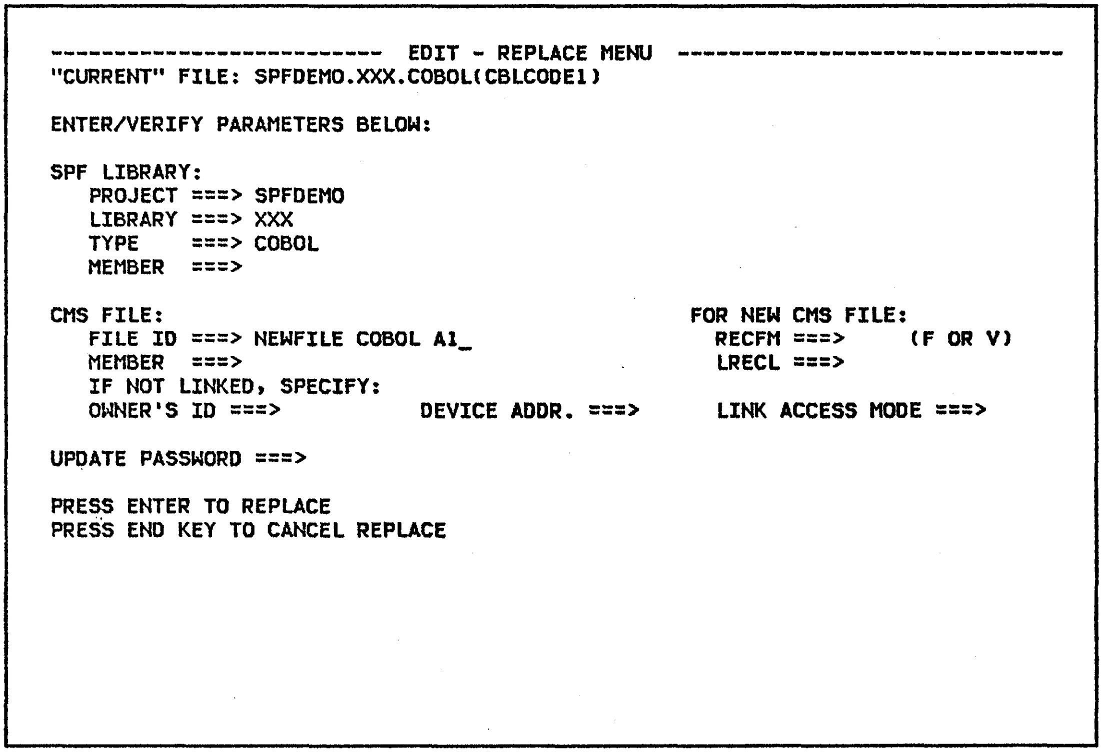
If the member-name operand is omitted on the CREATE or REPLACE primary command, the line command(s) designating the source of the operation must be entered before the file menu will be displayed.

Figure 39 shows an example of the REPLACE command. Member CBLCODE1 is being edited in SPF library SPFDEMO.XXX.COBOL. Lines 800 through 1700 are to be moved, as indicated with the "MM" line commands. The entire contents of file NEWFILE COBOL A1 are to be replaced with lines 800 through 1700.

Note: The data which is written to the destination file is renumbered if both NUMBER mode and AUTONUM are in effect. A source listing of the data is also recorded in the SPF list file for eventual printing, provided PRINT mode is in effect.



```
EDIT --- SPFDEMO.XXX.COBOL(CBLCODE1) - 01.01 ---------------- COLUMNS 007 078
COMMAND INPUT ===> REPLACE                                    SCROLL ===> HALF
000560      SELECT SYSPRINT
000570      ASSIGN TO UT-S-PRINT.
000600
000700  DATA DIVISION.
MM       FILE SECTION.
000900    FD  SYSPRINT
001000      LABEL RECORD IS OMITTED
001100      DATA RECORD IS PRINT-LINE.
001200    01  PRINT-LINE.
001300        02  LINE-FIELD    PICTURE X(121).
001400   WORKING-STORAGE SECTION.
001500    01  PRINT-VALUE       PICTURE X(121)
001600                               VALUE ' PRINT LINE PRODUCED BY SAMPL
MM_    -                                   'E CBL PROGRAM.'.
001800
001900  PROCEDURE DIVISION.
002000    BEGIN.
002100      OPEN OUTPUT SYSPRINT.
002200      WRITE PRINT-LINE FROM PRINT-VALUE.
002300      CLOSE SYSPRINT.
002400      STOP RUN.
****** *************************** BOTTOM OF DATA ****************************
```



```
-------------------------- EDIT - REPLACE MENU -----------------------------
"CURRENT" FILE: SPFDEMO.XXX.COBOL(CBLCODE1)

ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX
   TYPE    ===> COBOL
   MEMBER  ===>

CMS FILE:                                         FOR NEW CMS FILE:
   FILE ID ===> NEWFILE COBOL A1_                   RECFM ===>     (F OR V)
   MEMBER  ===>                                     LRECL ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>       LINK ACCESS MODE ===>

UPDATE PASSWORD ===>

PRESS ENTER TO REPLACE
PRESS END KEY TO CANCEL REPLACE
```

Figure 39. Edit - REPLACE command example

Nulls Control

The NULLS mode determines whether trailing blanks in each data field are written to the screen as blanks or nulls. The term "data field" normally refers to the 72 characters of data on each line. Using hardware tabs, however, each line can be split into multiple fields (see "Tabs Definition and Control").

NULLS mode may be turned on or off by using the NULLS command. Format:

NULLS [ON/OFF] [ALL]
Command abbreviation: NULL

The operands may be typed in any order. If the ON/OFF operand is omitted, ON is the default.

The ALL operand is valid only when NULLS mode is turned on. If ALL is typed, it causes all trailing blanks to be replaced with nulls and all-blank fields to be replaced with all-nulls. If ALL is omitted, one trailing blank is preserved (followed by nulls) and all-blank fields continue to contain blanks.

Blank characters (hexadecimal '40') and null characters (hexadecimal '00') both display as blanks. The reason for trailing nulls is to simplify use of the INSERT key on the IBM 3270 keyboard. This key may be used to insert characters on a line, providing the line contains trailing nulls. In addition to using the NULLS command, the user may create nulls at the end of a line via the ERASE EOF or DELETE key. Null characters are never stored in the data; they are always translated back to blanks.

Tabs Definition and Control

Three types of tabs are available: "software" tabs, "hardware" tabs, and "logical" tabs.

"Software" tabs are used by edit to reposition the cursor. Whenever the cursor is in the data portion of a line and the ENTER key is pressed, the cursor is moved to the next software tab position (providing software tabs have been defined).

"Hardware" tabs allow the use of the TAB FORWARD and TAB BACKWARD keys to move the cursor to user-defined column positions within the data. This is accomplished by inserting additional attribute bytes in each 72-character data area, which splits the data area into multiple fields. When the TAB FORWARD or TAB BACKWARD key is pressed, the cursor is moved to the screen position immediately following the next or previous attribute byte. There are always a least two attribute bytes on each edit line; one at the beginning of the line number field, and one at the beginning of the data.

The advantage of hardware tabs is that no program interrupt is generated when the TAB FORWARD and TAB BACKWARD keys are pressed. The disadvantage is that each attribute byte occupies a screen position and may not be overtyped. The attribute bytes display as blanks.

"Logical" tabs are used by edit to break up strings of data entered on a line and reposition the strings to user-defined tab positions. The beginning of each string is indicated by a user-designated special character.

Note: Tabs are not functional when using the text entry (TE) line command.

Use of tabs is controlled by two commands: The tabs line command which defines tab positions, and the TABS primary command which is used to turn TABS mode on and off.

Tabs Line Command

The tabs line command is used to define tab positions for all three types of tabs. To display the tab definition line, the characters "TABS" or "TAB" are entered in the line command area, overtyping the line numbers. When the ENTER key is pressed, the tab definition line is inserted at the designated position.

TABS - Display tab definition line
Command abbreviation: TAB

Software tab positions are defined by typing an underscore (_) or hyphen (-) at each desired column position on the tab definition line. Two or more consecutive underscores or hyphens may be typed to define a software tab field, which causes the cursor to be repositioned to the first non-blank character within the field. If the tab field contains all blanks, the cursor is positioned at the beginning of the field.

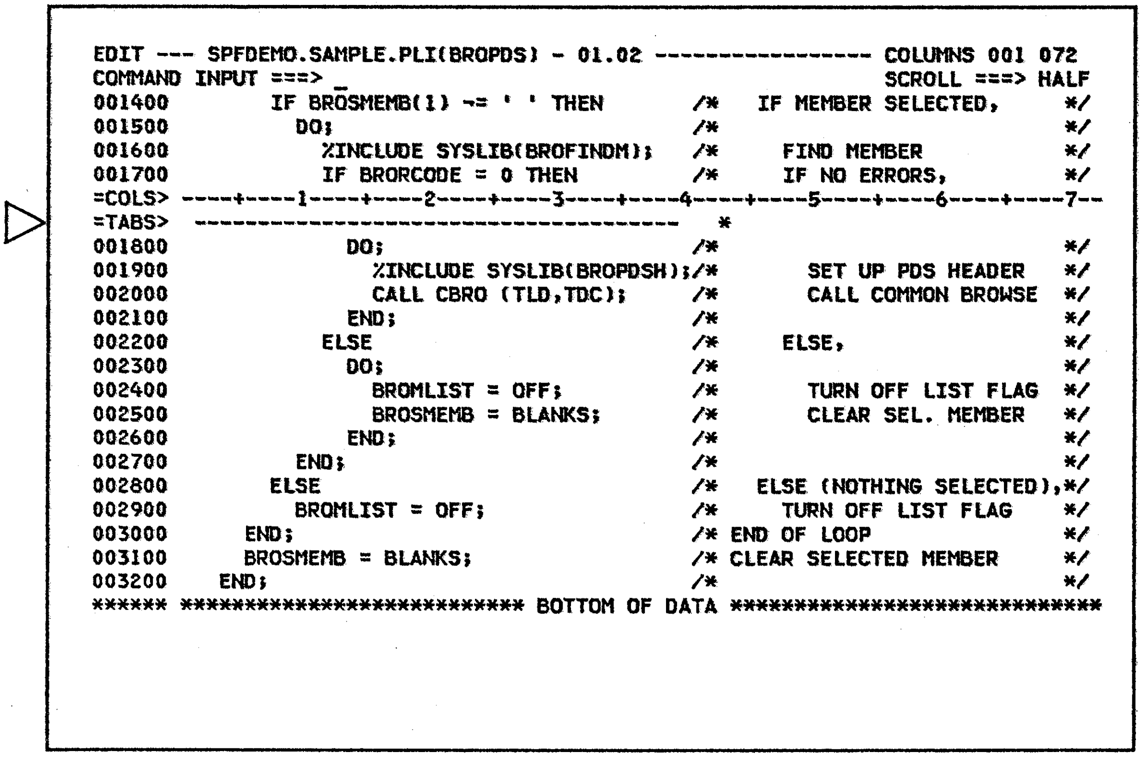
Hardware and logical tab positions are defined by typing an asterisk (*) at each desired column position on the tab definition line. The aterisks indicate the locations where the attribute bytes will be inserted (for hardware tabs) or the locations following which strings will be repositioned (for logical tabs).

Initially, the tab definition line contains all blanks. If the user displays the tab line and enters software tab definitions, they take effect immediately; each line contains a software tab or tab field at the designated column positions. Hardware and logical tab definitions do not take effect immediately. The asterisks define the column positions, but the insertion of attribute bytes (hardware tabs) or the repositioning of data strings (logical tabs) does not occur unless TABS mode is ON. The user may enter or leave TABS mode via the TABS ON/OFF primary command.

Figure 40 shows an example of the tab definition line displayed together with the column identification line. The tab line shows a software tab field extending from columns 2 through 39, and a hardware/logical tab definition at column 43.

The tab definition line may be removed from display by entering a "D" in the line command area, or by entering a RESET primary command. The tab definition line is never saved as part of the data.

The tab definitions remain in effect (even if not displayed) until they are again changed by the user. Tab definitions are retained in the current edit profile, and automatically used the next time the user edits the same type of data.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===> _                                          SCROLL ===> HALF
001400        IF BROSMEMB(1) ¬= ' ' THEN      /*   IF MEMBER SELECTED,     */
001500          DO;                           /*                             */
001600            %INCLUDE SYSLIB(BROFINDM);  /*     FIND MEMBER           */
001700            IF BRORCODE = 0 THEN        /*     IF NO ERRORS,         */
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
=TABS>  --------------------------------------   *
001800              DO;                       /*                             */
001900                %INCLUDE SYSLIB(BROPDSH);/*       SET UP PDS HEADER   */
002000                CALL CBRO (TLD,TDC);    /*       CALL COMMON BROWSE  */
002100              END;                      /*                             */
002200            ELSE                        /*     ELSE,                 */
002300              DO;                       /*                             */
002400                BROMLIST = OFF;         /*       TURN OFF LIST FLAG  */
002500                BROSMEMB = BLANKS;      /*       CLEAR SEL. MEMBER   */
002600              END;                      /*                             */
002700          END;                          /*                             */
002800        ELSE                            /*   ELSE (NOTHING SELECTED),*/
002900          BROMLIST = OFF;               /*     TURN OFF LIST FLAG    */
003000      END;                              /* END OF LOOP                 */
003100      BROSMEMB = BLANKS;                /* CLEAR SELECTED MEMBER     */
003200    END;                                /*                             */
****** *************************** BOTTOM OF DATA ****************************
```

Figure 40. Edit - tab definition line

TABS Primary Command

TABS mode may be turned on or off by using the TABS primary command. Format:

TABS [ON/OFF] [tab-character] [ALL]
Command abbreviation: TAB

The operands may be typed in any order. If the ON/OFF operand is omitted, ON is the default. The "tab-character" and "ALL" operands are valid only when TABS mode is turned on.

The "tab-character" operand consists of a single, non-alphameric (special) character. It defines the character to be interpreted as a logical tab when encountered on input. Example:

```
TABS ON $
```

If the user then enters the following information on a line:

```
$aaaaa$bbb$cccc
```

the data "aaaaa" will be repositioned after the first tab column (as defined by an "*" in the tab definition line), the "bbb" will be repositioned after the next tab column, etc., as follows:

```
TABS              *                   *                   *
                   aaaaa               bbb                 cccc
```

If the "tab-character" operand is not typed when TABS mode is turned on, hardware tabs are initiated by inserting attribute bytes at each tab position defined by an "*" in the tabs definition line. If the ALL operand is coded, an attribute byte is inserted at all user-defined positions on each line, overlaying any characters in those positions. If the ALL operand is omitted, an attribute byte is inserted at each user-defined position on each line only if that character position currently contains a blank or null.

Overlaying of characters with attribute bytes applies to the screen contents only; the attribute bytes are never recorded in the data. When TABS mode is turned off, the attribute bytes are removed and the overlaid character at each tab position is redisplayed.

In TABS mode, the user may temporarily cause the attribute bytes to be removed on a single line by blanking out the entire line command field or by placing the cursor directly under one of the attribute bytes and pressing the ENTER key. When the ENTER key is pressed again, the attribute bytes will be reinserted.

Profile Display and Control

The current edit profile may be displayed at any time by means of the PROFILE primary command. This command may also be used to define a new profile or switch to a different profile. Format:

PROFILE [name] [number]
Command abbreviations: PROF, PRO

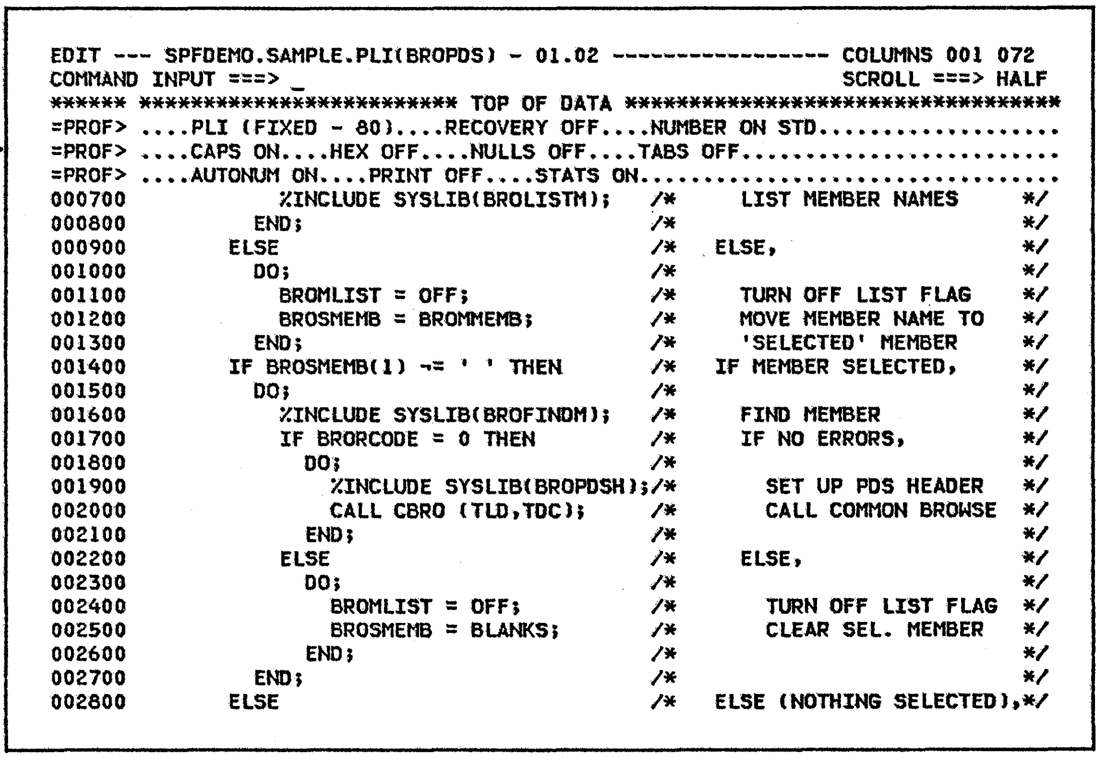
Both operands are optional, and may be coded in either order. The "name" operand consists of up to eight alphameric characters, the first of which must be alphabetic. The "number" operand consists of a single digit in the range 0 to 7, inclusive.

If neither operand is typed, the contents of the profile are displayed in the data area of the screen (Figure 41). The first three lines show the current mode settings. The remaining lines show the current contents of the MASK, TABS, and BOUNDS lines, together with the COLS positioning line. The MASK and TABS lines are not displayed if they contain all blanks, and the BOUNDS line is not displayed if it contains the default boundary positions.

If the "name" operand is typed, and if the user already has a profile of that name, edit immediately switches to the specified profile and displays it. (Display of the profile may be avoided by typing a zero for the "number" operand. See below.) If a profile of the specified name does not already exist, a new profile is defined with that name. The initial content of the new profile is the same at the profile which was in effect when the PROFILE command was entered.

The "number" operand controls the number of lines shown in the profile display. If the number 0 is typed, it prevents display of the profile. If the number 7 is typed, it forces display of all lines, even if the MASK and TABS are blanks and BOUNDS contains the defaults.

Use of profiles may also be controlled from the edit file menu. If the profile name field is left blank, the profile name defaults to the data "type" (SPF library type or CMS filetype). If a name is entered, it overrides the "type" qualifier. In either case, if a profile of that name currently exists, it is used. If it does not exist, a new profile is defined. The initial content of the new profile has the default mode settings (see "Edit Modes and Profiles"), all-blank MASK and TABS, and default BOUNDS.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===> _                                          SCROLL ===> HALF
****** ************************* TOP OF DATA **********************************
=PROF> ....PLI (FIXED - 80)....RECOVERY OFF....NUMBER ON STD..................
=PROF> ....CAPS ON....HEX OFF....NULLS OFF....TABS OFF........................
=PROF> ....AUTONUM ON....PRINT OFF....STATS ON................................
000700            %INCLUDE SYSLIB(BROLISTM);   /*     LIST MEMBER NAMES     */
000800          END;                           /*                           */
000900        ELSE                             /*   ELSE,                   */
001000          DO;                            /*                           */
001100            BROMLIST = OFF;              /*     TURN OFF LIST FLAG    */
001200            BROSMEMB = BROMMEMB;         /*     MOVE MEMBER NAME TO   */
001300          END;                           /*     'SELECTED' MEMBER     */
001400        IF BROSMEMB(1) ¬= ' ' THEN       /*   IF MEMBER SELECTED,     */
001500          DO;                            /*                           */
001600            %INCLUDE SYSLIB(BROFINDM);   /*     FIND MEMBER           */
001700            IF BRORCODE = 0 THEN         /*     IF NO ERRORS,         */
001800              DO;                        /*                           */
001900                %INCLUDE SYSLIB(BROPDSH);/*       SET UP PDS HEADER   */
002000                CALL CBRO (TLD,TDC);     /*       CALL COMMON BROWSE  */
002100              END;                       /*                           */
002200            ELSE                         /*     ELSE,                 */
002300              DO;                        /*                           */
002400                BROMLIST = OFF;          /*       TURN OFF LIST FLAG  */
002500                BROSMEMB = BLANKS;       /*       CLEAR SEL. MEMBER   */
002600              END;                       /*                           */
002700          END;                           /*                           */
002800        ELSE                             /*   ELSE (NOTHING SELECTED),*/
```

Figure 41. Edit - profile display

Hexadecimal Display

When edit is operating in HEX mode, three lines are displayed for each source line. The first line shows the data in standard character form. The following two lines show the same data in hexadecimal representation. See Figure 42.

Any invalid (non-displayable) characters may be changed by overtyping the hexadecimal representations. Any valid character may be changed by overtyping either the standard character representations or the hexadecimal representations.

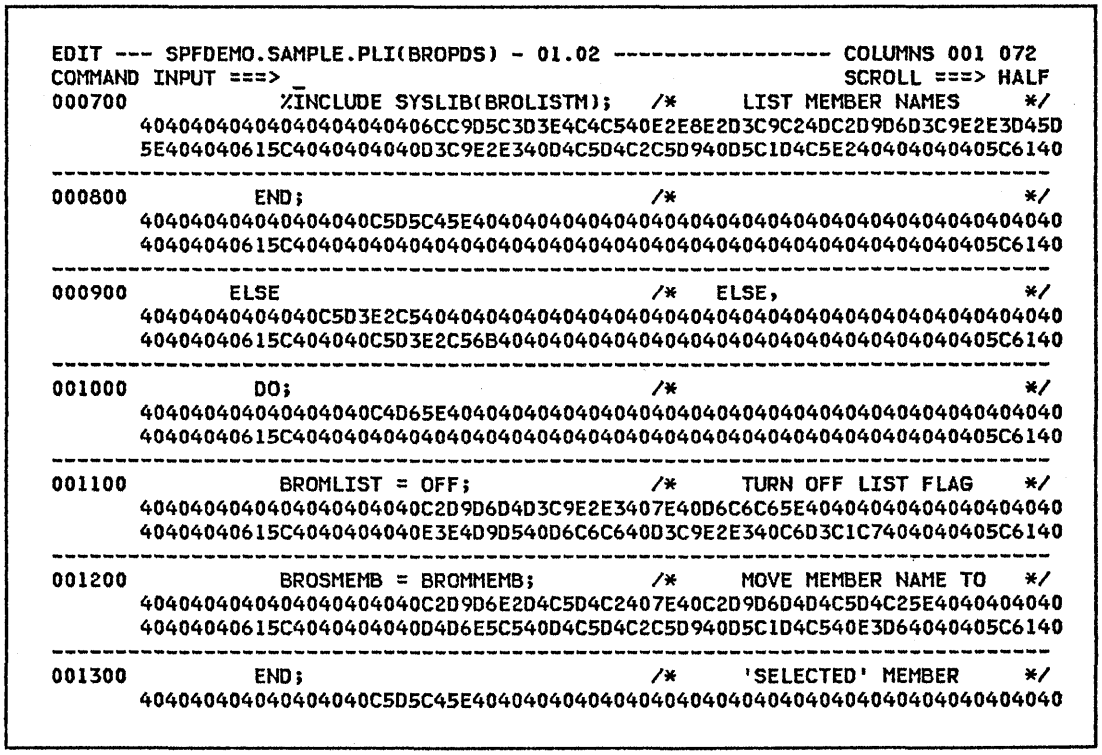
Note: The FIND and CHANGE commands may also be used to find and change invalid characters or any specific hex character, regardless of the setting of HEX mode. See discussion of picture strings and hex strings under FIND and CHANGE commands.

HEX mode may be turned on or off by using the HEX command. Format:

```
   HEX  [ON/OFF]  [VERT/DATA]
```

The operands may be typed in any order. If the ON/OFF operand is omitted, ON is the default.

The VERT (vertical) and DATA operands are valid only when HEX mode is turned on. VERT causes the hexadecimal representations to be displayed vertically (two rows per byte) under each character. DATA causes the hexadecimal representations to be displayed as a string of hex characters (two per byte), as shown in Figure 42. Since the hex string is twice as long as the data string, it consumes two rows. If this operand is omitted, VERT is the default.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND INPUT ===> _                                          SCROLL ===> HALF
000700            %INCLUDE SYSLIB(BROLISTM);   /*      LIST MEMBER NAMES     */
       4040404040404040404040406CC9D5C3D3E4C4C540E2E8E2D3C9C24DC2D9D6D3C9E2E3D45D
       5E404040615C4040404040D3C9E2E340D4C5D4C2C5D940D5C1D4C5E240404040405C6140
-----------------------------------------------------------------------------
000800          END;                           /*                            */
       40404040404040C5D5C45E40404040404040404040404040404040404040404040404040
       40404040615C4040404040404040404040404040404040404040404040404040405C6140
-----------------------------------------------------------------------------
000900        ELSE                             /*   ELSE,                    */
       404040404040C5D3E2C54040404040404040404040404040404040404040404040404040
       40404040615C404040C5D3E2C56B404040404040404040404040404040404040405C6140
-----------------------------------------------------------------------------
001000          DO;                            /*                            */
       40404040404040C4D65E4040404040404040404040404040404040404040404040404040
       40404040615C4040404040404040404040404040404040404040404040404040405C6140
-----------------------------------------------------------------------------
001100            BROMLIST = OFF;              /*      TURN OFF LIST FLAG    */
       4040404040404040404040C2D9D6D4D3C9E2E3407E40D6C6C65E40404040404040404040
       40404040615C4040404040E3E4D9D540D6C6C640D3C9E2E340C6D3C1C7404040405C6140
-----------------------------------------------------------------------------
001200            BROSMEMB = BROMMEMB;         /*      MOVE MEMBER NAME TO   */
       4040404040404040404040C2D9D6E2D4C5D4C2407E40C2D9D6D4D4C5D4C25E4040404040
       40404040615C4040404040D4D6E5C540D4C5D4C2C5D940D5C1D4C540E3D64040405C6140
-----------------------------------------------------------------------------
001300          END;                           /*      'SELECTED' MEMBER     */
       40404040404040C5D5C45E40404040404040404040404040404040404040404040404040
```

Figure 42. Edit - hexadecimal display, data

TEXT PREPARATION FEATURES

The following commands are described in the section:

Primary command:	CAPS	
Line commands:	TE	(text entry)
	TS	(text split)
	TF	(text flow)
	O	(overlay)

Character Translation

If CAPS mode is in on, alphabetic data entered at the terminal is automatically translated to upper case during edit operations. If CAPS mode is off, alphabetic data is left "as is". CAPS mode is normally on except when documentation (text) is being edited.

CAPS mode also determines whether character strings, entered as operands of the FIND and CHANGE commands, are translated to upper case. Note that the keyword operands and all menu parameters are always translated to upper case regardless of the current setting of CAPS mode.

CAPS mode may be turned on or off by using the CAPS command. Format:

```
    CAPS  [ON/OFF]
```

If the operand is omitted, ON is the default.

Whenever data is fetched for editing, SPF will initialize the setting of CAPS mode based on whether the data does or does not contain lower case alphabetics. If the initial setting of CAPS mode differs from the previous setting in the profile, a message will be displayed indicating that edit has changed the mode. For new files or members, the initial setting of CAPS mode is unchanged from whatever is in the profile. The default for a new profile is CAPS mode ON.

Text Entry and Edit

All of the edit line and primary commands may be used when entering and editing text information. Three additional line commands are provided specifically for this purpose:

TE - Text Entry
TS - Text Split
TF - Text Flow

These commands are described in the following sections.

Text Entry

The text entry command is intended for bulk entry of text using power typing techniques. When the characters "TE" are entered in a line command area, blank lines are inserted starting with the line following the "TE". A number may follow the "TE" to indicate the number of blank lines to be inserted. If the number is omitted, sufficient blank lines are inserted to fill the screen from the line following the "TE" to the end of the screen.

```
TE   - Text entry (to end of screen)
TEn  - Text entry "n" lines
```

The user may type information on the blank lines without regard to line end. The line command area on the blank lines is protected, and automatic cursor skip is provided from the end of one line to the first data position on the next line. When the ENTER key is pressed, the new information is automatically reflowed to fit within the currently defined boundaries (see BOUNDS line command), and any unused blank lines at the end of the new text are deleted.

If insufficient blank lines have been generated, the keyboard will lock when the user attempts to type beyond the last character position of the last blank line. A vertical bar (|) will be displayed above the cursor at the locked position. To generate more blank lines, press the RESET key to unlock the keyboard and then press the ENTER key.

Multiple paragraphs may be entered by using either of the following techniques:

- leave a blank line between paragraphs, or
- indent the start of a new paragraph with three or more blank (space) characters.

Note: Tab definitions are not functional during text entry.

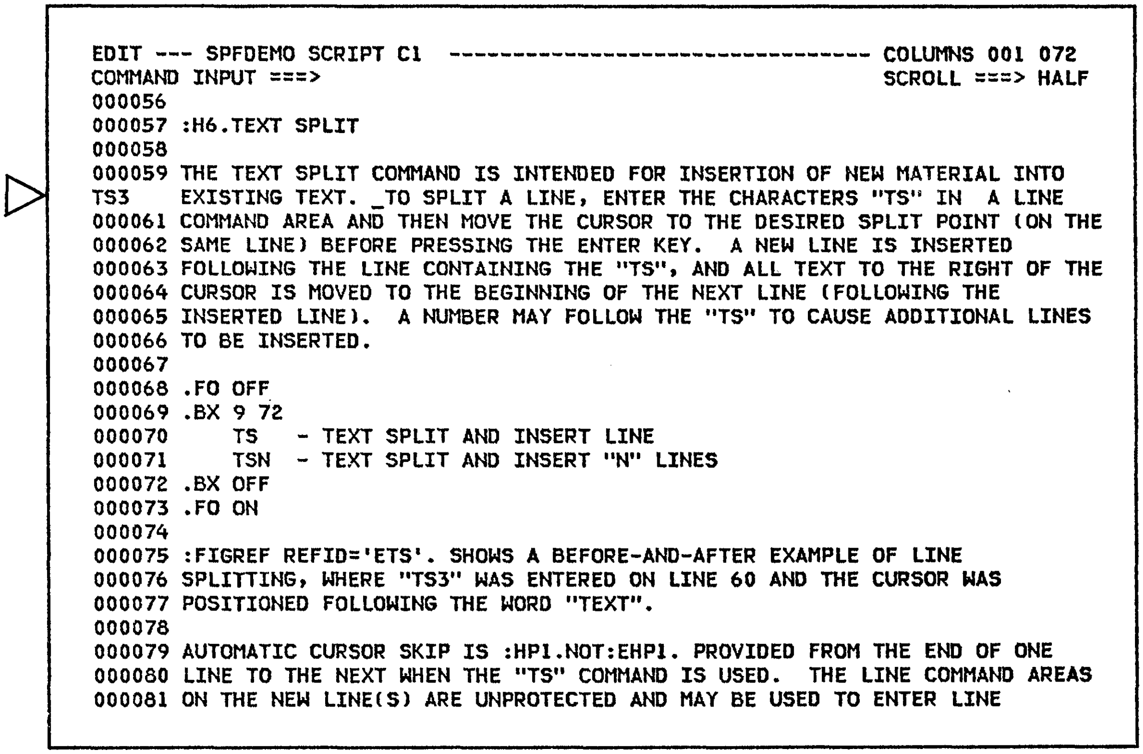
Text Split

The text split command is intended for insertion of new material into existing text. To split a line, enter the characters "TS" in a line command area and then move the cursor to the desired split point (on the same line) before pressing the ENTER key. A new line is inserted following the line containing the "TS", and all text to the right of the cursor is moved to the beginning of the next line (following the inserted line). A number may follow the "TS" to cause additional lines to be inserted.

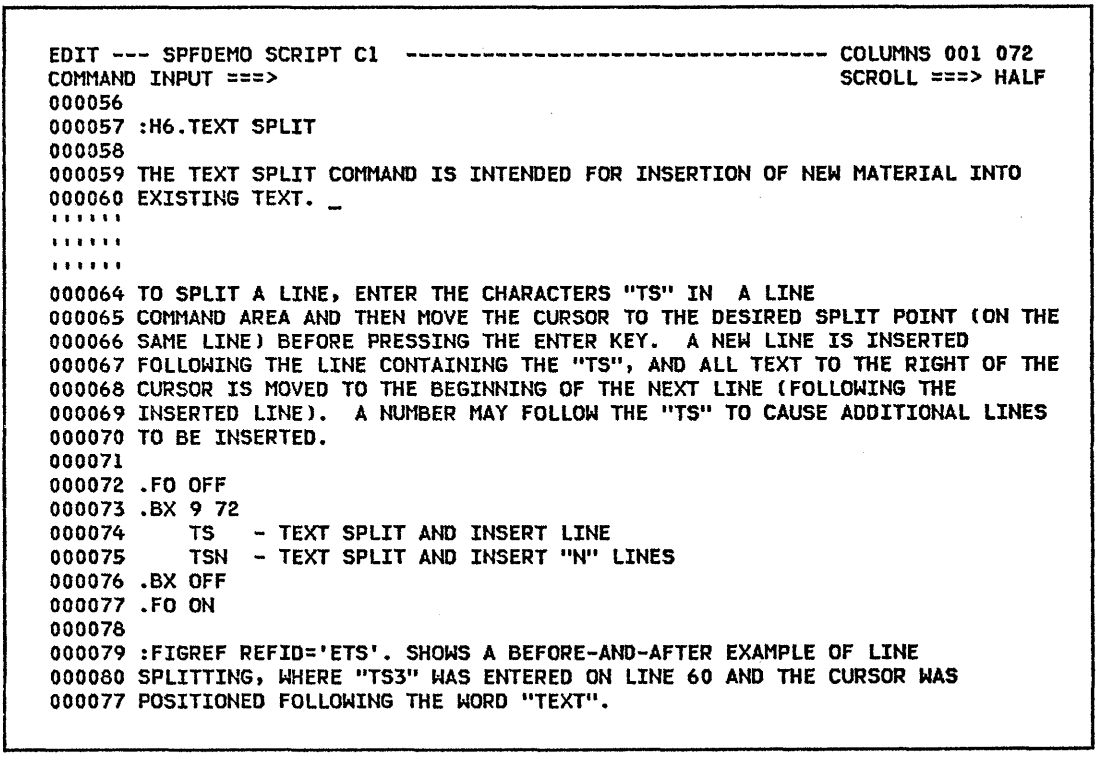
```
TS   - Text split and insert line
TSn  - Text split and insert "n" lines
```

Figure 43 shows a before-and-after example of line splitting, where "TS3" was entered on line 60 and the cursor was positioned following the word "text".

Automatic cursor skip is not provided from the end of one line to the next when the "TS" command is used. The line command areas on the new line(s) are unprotected and may be used to enter line commands, as usual.



```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000056
000057 :H6.TEXT SPLIT
000058
000059 THE TEXT SPLIT COMMAND IS INTENDED FOR INSERTION OF NEW MATERIAL INTO
TS3    EXISTING TEXT. _TO SPLIT A LINE, ENTER THE CHARACTERS "TS" IN  A LINE
000061 COMMAND AREA AND THEN MOVE THE CURSOR TO THE DESIRED SPLIT POINT (ON THE
000062 SAME LINE) BEFORE PRESSING THE ENTER KEY.  A NEW LINE IS INSERTED
000063 FOLLOWING THE LINE CONTAINING THE "TS", AND ALL TEXT TO THE RIGHT OF THE
000064 CURSOR IS MOVED TO THE BEGINNING OF THE NEXT LINE (FOLLOWING THE
000065 INSERTED LINE).  A NUMBER MAY FOLLOW THE "TS" TO CAUSE ADDITIONAL LINES
000066 TO BE INSERTED.
000067
000068 .FO OFF
000069 .BX 9 72
000070     TS    - TEXT SPLIT AND INSERT LINE
000071     TSN   - TEXT SPLIT AND INSERT "N" LINES
000072 .BX OFF
000073 .FO ON
000074
000075 :FIGREF REFID='ETS'. SHOWS A BEFORE-AND-AFTER EXAMPLE OF LINE
000076 SPLITTING, WHERE "TS3" WAS ENTERED ON LINE 60 AND THE CURSOR WAS
000077 POSITIONED FOLLOWING THE WORD "TEXT".
000078
000079 AUTOMATIC CURSOR SKIP IS :HP1.NOT:EHP1. PROVIDED FROM THE END OF ONE
000080 LINE TO THE NEXT WHEN THE "TS" COMMAND IS USED.  THE LINE COMMAND AREAS
000081 ON THE NEW LINE(S) ARE UNPROTECTED AND MAY BE USED TO ENTER LINE
```

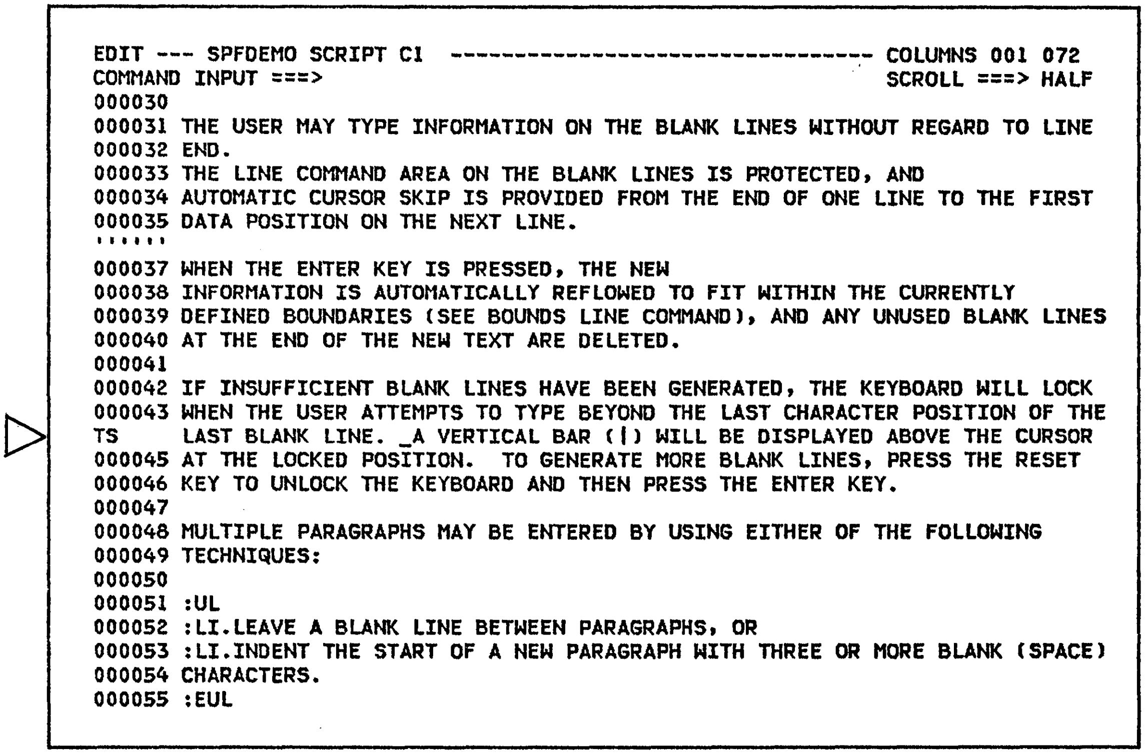


```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000056
000057 :H6.TEXT SPLIT
000058
000059 THE TEXT SPLIT COMMAND IS INTENDED FOR INSERTION OF NEW MATERIAL INTO
000060 EXISTING TEXT. _
''''''
''''''
''''''
000064 TO SPLIT A LINE, ENTER THE CHARACTERS "TS" IN  A LINE
000065 COMMAND AREA AND THEN MOVE THE CURSOR TO THE DESIRED SPLIT POINT (ON THE
000066 SAME LINE) BEFORE PRESSING THE ENTER KEY.  A NEW LINE IS INSERTED
000067 FOLLOWING THE LINE CONTAINING THE "TS", AND ALL TEXT TO THE RIGHT OF THE
000068 CURSOR IS MOVED TO THE BEGINNING OF THE NEXT LINE (FOLLOWING THE
000069 INSERTED LINE).  A NUMBER MAY FOLLOW THE "TS" TO CAUSE ADDITIONAL LINES
000070 TO BE INSERTED.
000071
000072 .FO OFF
000073 .BX 9 72
000074     TS    - TEXT SPLIT AND INSERT LINE
000075     TSN   - TEXT SPLIT AND INSERT "N" LINES
000076 .BX OFF
000077 .FO ON
000078
000079 :FIGREF REFID='ETS'. SHOWS A BEFORE-AND-AFTER EXAMPLE OF LINE
000080 SPLITTING, WHERE "TS3" WAS ENTERED ON LINE 60 AND THE CURSOR WAS
000077 POSITIONED FOLLOWING THE WORD "TEXT".
```

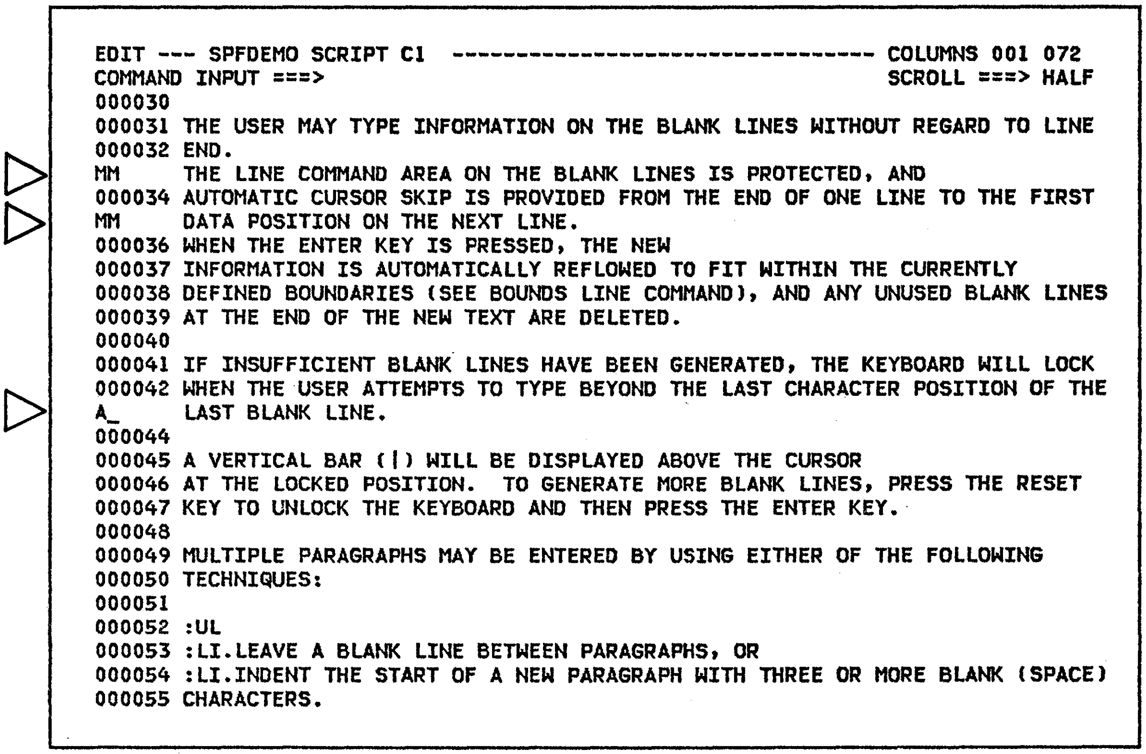
Figure 43. Edit - text split example

The text split command may also be used in conjunction with other line commands to move or copy text. Use "TS" at the end points of the material to be moved or copied (i.e., to break it out so that is is wholly contained on one or more lines). In addition, use "TS" at the insertion point to open up space. Then use the standard move ("M" or "MM") or copy ("C" or "CC") line command together with the after ("A") or before ("B") line command.

Figure 44 shows an example of a sentence to be moved from one paragraph to another. In the first part of the figure, the sentence on lines 33-35 has already been split out by using the "TS" command twice (once with the cursor at the beginning of the sentence, and again at the end of the sentence). Another "TS" command is now entered on line 44 to split the second paragraph at the destination point for the move. The second part of the figure shows the use of "MM" and "A" to accomplish the move.



```
EDIT --- SPFDEMO SCRIPT C1 -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                          SCROLL ===> HALF
000030
000031 THE USER MAY TYPE INFORMATION ON THE BLANK LINES WITHOUT REGARD TO LINE
000032 END.
000033 THE LINE COMMAND AREA ON THE BLANK LINES IS PROTECTED, AND
000034 AUTOMATIC CURSOR SKIP IS PROVIDED FROM THE END OF ONE LINE TO THE FIRST
000035 DATA POSITION ON THE NEXT LINE.
''''''
000037 WHEN THE ENTER KEY IS PRESSED, THE NEW
000038 INFORMATION IS AUTOMATICALLY REFLOWED TO FIT WITHIN THE CURRENTLY
000039 DEFINED BOUNDARIES (SEE BOUNDS LINE COMMAND), AND ANY UNUSED BLANK LINES
000040 AT THE END OF THE NEW TEXT ARE DELETED.
000041
000042 IF INSUFFICIENT BLANK LINES HAVE BEEN GENERATED, THE KEYBOARD WILL LOCK
000043 WHEN THE USER ATTEMPTS TO TYPE BEYOND THE LAST CHARACTER POSITION OF THE
TS     LAST BLANK LINE. _A VERTICAL BAR (|) WILL BE DISPLAYED ABOVE THE CURSOR
000045 AT THE LOCKED POSITION.  TO GENERATE MORE BLANK LINES, PRESS THE RESET
000046 KEY TO UNLOCK THE KEYBOARD AND THEN PRESS THE ENTER KEY.
000047
000048 MULTIPLE PARAGRAPHS MAY BE ENTERED BY USING EITHER OF THE FOLLOWING
000049 TECHNIQUES:
000050
000051 :UL
000052 :LI.LEAVE A BLANK LINE BETWEEN PARAGRAPHS, OR
000053 :LI.INDENT THE START OF A NEW PARAGRAPH WITH THREE OR MORE BLANK (SPACE)
000054 CHARACTERS.
000055 :EUL
```



```
EDIT --- SPFDEMO SCRIPT C1 -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                          SCROLL ===> HALF
000030
000031 THE USER MAY TYPE INFORMATION ON THE BLANK LINES WITHOUT REGARD TO LINE
000032 END.
MM     THE LINE COMMAND AREA ON THE BLANK LINES IS PROTECTED, AND
000034 AUTOMATIC CURSOR SKIP IS PROVIDED FROM THE END OF ONE LINE TO THE FIRST
MM     DATA POSITION ON THE NEXT LINE.
000036 WHEN THE ENTER KEY IS PRESSED, THE NEW
000037 INFORMATION IS AUTOMATICALLY REFLOWED TO FIT WITHIN THE CURRENTLY
000038 DEFINED BOUNDARIES (SEE BOUNDS LINE COMMAND), AND ANY UNUSED BLANK LINES
000039 AT THE END OF THE NEW TEXT ARE DELETED.
000040
000041 IF INSUFFICIENT BLANK LINES HAVE BEEN GENERATED, THE KEYBOARD WILL LOCK
000042 WHEN THE USER ATTEMPTS TO TYPE BEYOND THE LAST CHARACTER POSITION OF THE
A_     LAST BLANK LINE.
000044
000045 A VERTICAL BAR (|) WILL BE DISPLAYED ABOVE THE CURSOR
000046 AT THE LOCKED POSITION.  TO GENERATE MORE BLANK LINES, PRESS THE RESET
000047 KEY TO UNLOCK THE KEYBOARD AND THEN PRESS THE ENTER KEY.
000048
000049 MULTIPLE PARAGRAPHS MAY BE ENTERED BY USING EITHER OF THE FOLLOWING
000050 TECHNIQUES:
000051
000052 :UL
000053 :LI.LEAVE A BLANK LINE BETWEEN PARAGRAPHS, OR
000054 :LI.INDENT THE START OF A NEW PARAGRAPH WITH THREE OR MORE BLANK (SPACE)
000055 CHARACTERS.
```

Figure 44. Edit - move sentence example

Text Flow

The text flow command may be used to reflow paragraphs following deletions, insertions, splitting, etc. When the characters "TF" are entered in a line command area, the text is reflowed from the beginning of that line to the end of the paragraph.

```
TF   - Text flow (between current bounds)
TFn  - Text flow from left bound to column "n"
```

The reflow operation will remove trailing blanks on a line by pulling in material from the following line. It will not, however, remove embedded blanks within a line. Accordingly, if one or more words in a line are to be deleted, the 3270 DELETE key should be used (rather than overtyping the words with blanks).

The text to be flowed is taken from within the currently defined column boundaries (see BOUNDS line command). Any text outside the bounds does not participate in the flow operation. The reflowed text is also positioned within the current bounds. If the original text was indented from the left boundary, the indentation is preserved.

Each line of the reflowed text normally extends to the right boundary position. However, the text may be "squeezed up" by entering a number after the "TF" command. The number specifies the right-most column postion for the reflowed text. It must be less than the current setting of the right bound (otherwise, it is ignored).

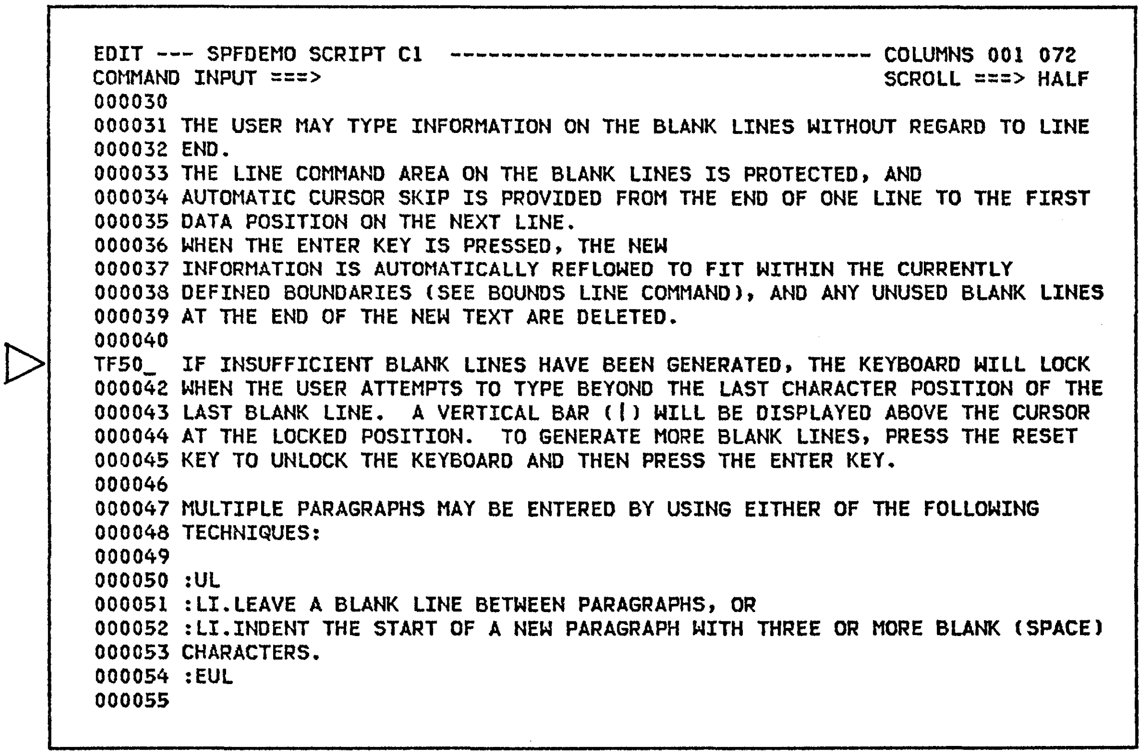
A before and after example of text flow is shown in Figure 45. The bounds are set at 1 and 72. A "TF50" command is entered on line 41. All text between 1 and 72 participates in the flow, but the results are contained within columns 1 and 50.

Use of Program Function Keys for Text Commands

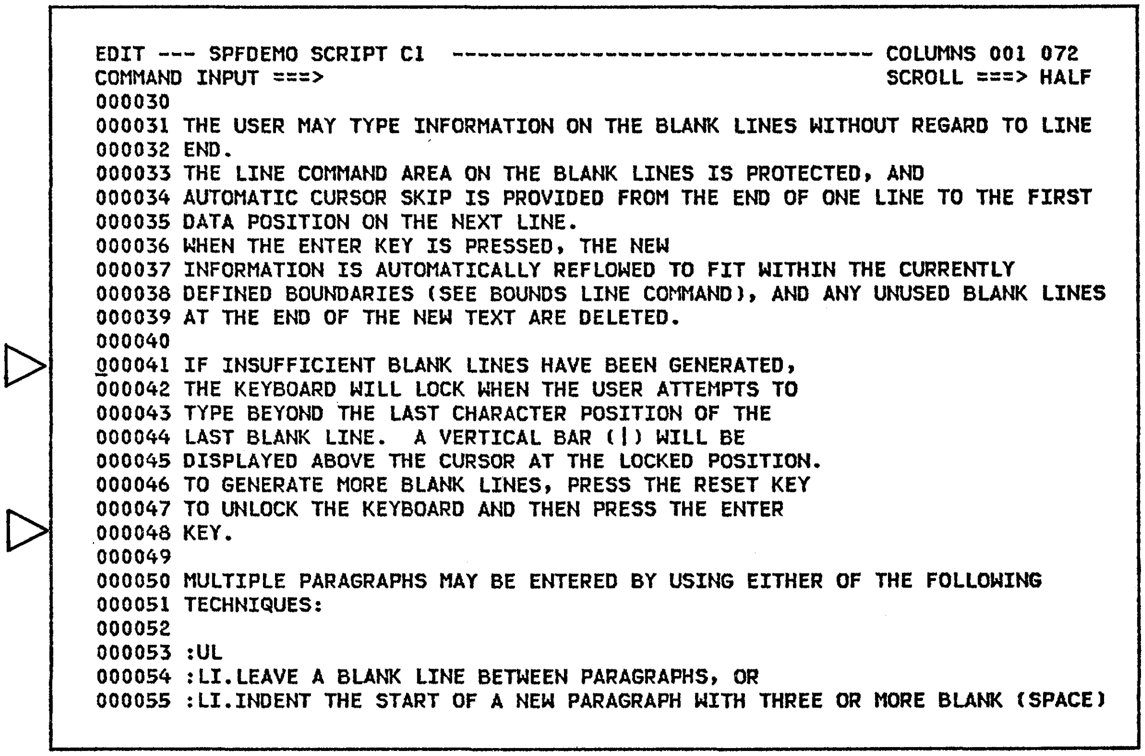
Whenever heavy use is being made of the text line commands, it is strongly recommended that both the text split and text flow commands be assigned to program function (PF) keys. For users of 12-key terminals, infrequently used PF keys such as Shift Right and Shift Left may be reassigned to these functions. Use SPF Parms (option 0.3) to reassign the keys. Example:

```
PF10 ===> :TS
PF11 ===> :TF
```

With these key definitions, a text split may be accomplished by moving the cursor to the desired split point (within a line) and pressing PF10. After the new material has been typed, PF11 may be pressed to reflow the text from the line containing the cursor to the end of the paragraph.



```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000030
000031 THE USER MAY TYPE INFORMATION ON THE BLANK LINES WITHOUT REGARD TO LINE
000032 END.
000033 THE LINE COMMAND AREA ON THE BLANK LINES IS PROTECTED, AND
000034 AUTOMATIC CURSOR SKIP IS PROVIDED FROM THE END OF ONE LINE TO THE FIRST
000035 DATA POSITION ON THE NEXT LINE.
000036 WHEN THE ENTER KEY IS PRESSED, THE NEW
000037 INFORMATION IS AUTOMATICALLY REFLOWED TO FIT WITHIN THE CURRENTLY
000038 DEFINED BOUNDARIES (SEE BOUNDS LINE COMMAND), AND ANY UNUSED BLANK LINES
000039 AT THE END OF THE NEW TEXT ARE DELETED.
000040
TF50_  IF INSUFFICIENT BLANK LINES HAVE BEEN GENERATED, THE KEYBOARD WILL LOCK
000042 WHEN THE USER ATTEMPTS TO TYPE BEYOND THE LAST CHARACTER POSITION OF THE
000043 LAST BLANK LINE.  A VERTICAL BAR (|) WILL BE DISPLAYED ABOVE THE CURSOR
000044 AT THE LOCKED POSITION.  TO GENERATE MORE BLANK LINES, PRESS THE RESET
000045 KEY TO UNLOCK THE KEYBOARD AND THEN PRESS THE ENTER KEY.
000046
000047 MULTIPLE PARAGRAPHS MAY BE ENTERED BY USING EITHER OF THE FOLLOWING
000048 TECHNIQUES:
000049
000050 :UL
000051 :LI.LEAVE A BLANK LINE BETWEEN PARAGRAPHS, OR
000052 :LI.INDENT THE START OF A NEW PARAGRAPH WITH THREE OR MORE BLANK (SPACE)
000053 CHARACTERS.
000054 :EUL
000055
```



```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000030
000031 THE USER MAY TYPE INFORMATION ON THE BLANK LINES WITHOUT REGARD TO LINE
000032 END.
000033 THE LINE COMMAND AREA ON THE BLANK LINES IS PROTECTED, AND
000034 AUTOMATIC CURSOR SKIP IS PROVIDED FROM THE END OF ONE LINE TO THE FIRST
000035 DATA POSITION ON THE NEXT LINE.
000036 WHEN THE ENTER KEY IS PRESSED, THE NEW
000037 INFORMATION IS AUTOMATICALLY REFLOWED TO FIT WITHIN THE CURRENTLY
000038 DEFINED BOUNDARIES (SEE BOUNDS LINE COMMAND), AND ANY UNUSED BLANK LINES
000039 AT THE END OF THE NEW TEXT ARE DELETED.
000040
000041 IF INSUFFICIENT BLANK LINES HAVE BEEN GENERATED,
000042 THE KEYBOARD WILL LOCK WHEN THE USER ATTEMPTS TO
000043 TYPE BEYOND THE LAST CHARACTER POSITION OF THE
000044 LAST BLANK LINE.  A VERTICAL BAR (|) WILL BE
000045 DISPLAYED ABOVE THE CURSOR AT THE LOCKED POSITION.
000046 TO GENERATE MORE BLANK LINES, PRESS THE RESET KEY
000047 TO UNLOCK THE KEYBOARD AND THEN PRESS THE ENTER
000048 KEY.
000049
000050 MULTIPLE PARAGRAPHS MAY BE ENTERED BY USING EITHER OF THE FOLLOWING
000051 TECHNIQUES:
000052
000053 :UL
000054 :LI.LEAVE A BLANK LINE BETWEEN PARAGRAPHS, OR
000055 :LI.INDENT THE START OF A NEW PARAGRAPH WITH THREE OR MORE BLANK (SPACE)
```

Figure 45. Edit - text flow example

Overlaying Lines

The overlay line ("O" or "OO") command allows a list of items to be rearranged in multi-column (tabular) format. The overlay command may be used with the move line ("M" or "MM") or copy line ("C" or "CC") line commands. It is used in place of the after ("A") command.

The character "O" may be entered in a line command area to specify the line over which data is to be moved or copied. A number may follow the "O" to indicate the number of lines to be overlayed. A block of lines to be overlayed may be specified by entering "OO" on the first and last lines of the block.

O	- Overlay line
On	- Overlay "n" lines
OO	- Overlay block of lines

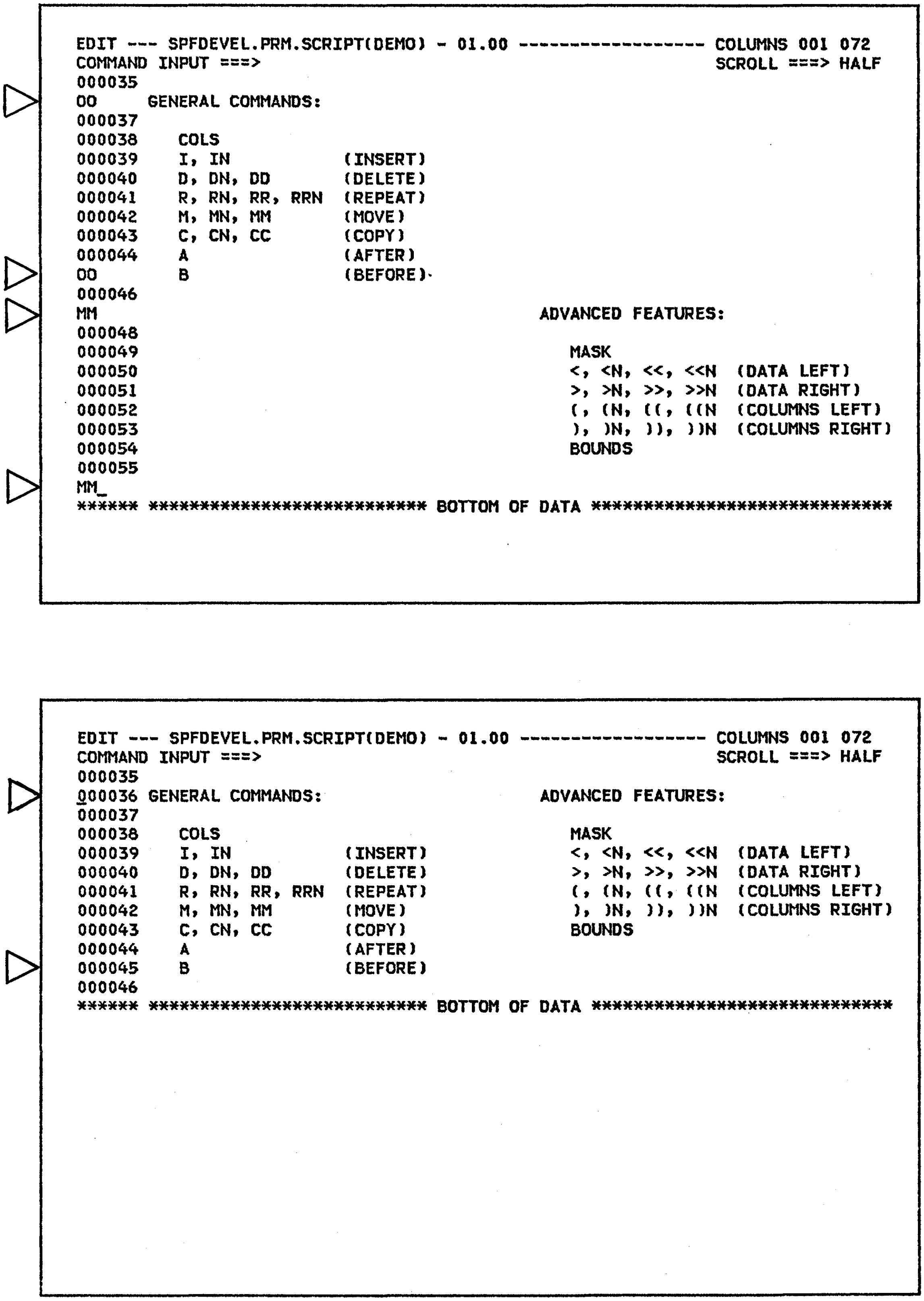
Blank characters in the receiving line(s), specified with "O" or "OO", are overlayed with corresponding characters from the source line(s), specified with "M", "MM", "C", or "CC". Non-blank characters are not overlayed.

Only those characters which are within the current column boundaries participate in the overlay operation (see description of BOUNDS command).

The number of source and receiving lines need not be the same. If there are more receiving lines, the source lines are repeated until the receiving lines are used up. If there are more source lines, the extra source lines are ignored.

Only data lines participate in the overlay operation. Special lines such as MASK, TABS, BOUNDS, and COLS are ignored as either source or receiving lines.

A before and after example of overlaying lines is shown in Figure 46. Generally, a list such as that shown in the figure would be maintained and edited as a single column (left adjusted). Prior to the overlay, portions of the list would be right-shifted by the appropriate amounts to overlay in the multi-column format.



```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.00 ------------------ COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000035
OO     GENERAL COMMANDS:
000037
000038    COLS
000039    I, IN            (INSERT)
000040    D, DN, DD        (DELETE)
000041    R, RN, RR, RRN   (REPEAT)
000042    M, MN, MM        (MOVE)
000043    C, CN, CC        (COPY)
000044    A                (AFTER)
OO        B                (BEFORE)
000046
MM                                    ADVANCED FEATURES:
000048
000049                                   MASK
000050                                   <, <N, <<, <<N  (DATA LEFT)
000051                                   >, >N, >>, >>N  (DATA RIGHT)
000052                                   (, (N, ((, ((N  (COLUMNS LEFT)
000053                                   ), )N, )), ))N  (COLUMNS RIGHT)
000054                                   BOUNDS
000055
MM_
****** *************************** BOTTOM OF DATA ****************************
```

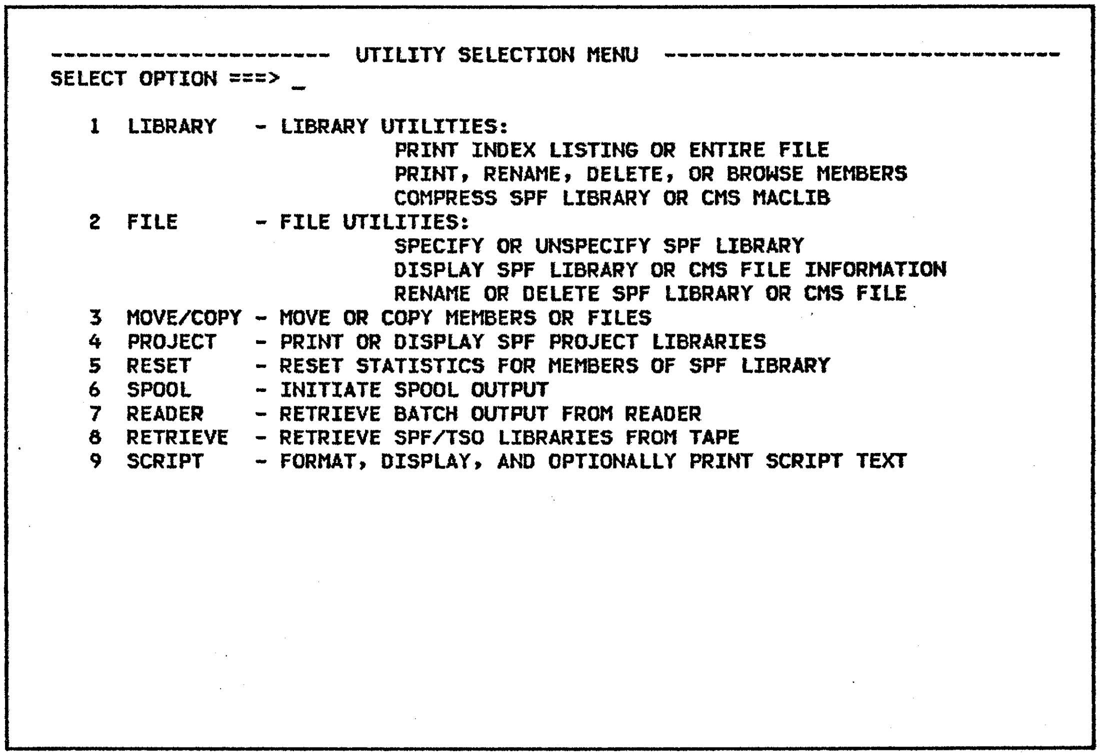
```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.00 ------------------ COLUMNS 001 072
COMMAND INPUT ===>                                            SCROLL ===> HALF
000035
000036 GENERAL COMMANDS:              ADVANCED FEATURES:
000037
000038    COLS                           MASK
000039    I, IN            (INSERT)      <, <N, <<, <<N  (DATA LEFT)
000040    D, DN, DD        (DELETE)      >, >N, >>, >>N  (DATA RIGHT)
000041    R, RN, RR, RRN   (REPEAT)      (, (N, ((, ((N  (COLUMNS LEFT)
000042    M, MN, MM        (MOVE)        ), )N, )), ))N  (COLUMNS RIGHT)
000043    C, CN, CC        (COPY)        BOUNDS
000044    A                (AFTER)
000045    B                (BEFORE)
000046
****** *************************** BOTTOM OF DATA ****************************
```

Figure 46. Edit - overlay line example

UTILITIES (OPTION 3)

The utility option provides a variety of functions for: SPF library and file maintenance, moving and copying data, printing or displaying SPF project listings, resetting SPF library statistics, initiating spool output, retrieving data from the virtual reader, retrieving SPF libraries (via tape) from a TSO system, and formatting SCRIPT/VS documentation.

When the utility option is selected (primary option 3), a secondary selection menu is displayed (Figure 47). It is possible to bypass the secondary menu by entering two numbers, separated with a decimal point, on the primary option menu. For example, entering "3.2" on the primary menu has the same effect as entering "3" on the primary menu and "2" on the secondary menu.



```
---------------------- UTILITY SELECTION MENU ------------------------------
SELECT OPTION ===> _

   1  LIBRARY   - LIBRARY UTILITIES:
                           PRINT INDEX LISTING OR ENTIRE FILE
                           PRINT, RENAME, DELETE, OR BROWSE MEMBERS
                           COMPRESS SPF LIBRARY OR CMS MACLIB
   2  FILE      - FILE UTILITIES:
                           SPECIFY OR UNSPECIFY SPF LIBRARY
                           DISPLAY SPF LIBRARY OR CMS FILE INFORMATION
                           RENAME OR DELETE SPF LIBRARY OR CMS FILE
   3  MOVE/COPY - MOVE OR COPY MEMBERS OR FILES
   4  PROJECT   - PRINT OR DISPLAY SPF PROJECT LIBRARIES
   5  RESET     - RESET STATISTICS FOR MEMBERS OF SPF LIBRARY
   6  SPOOL     - INITIATE SPOOL OUTPUT
   7  READER    - RETRIEVE BATCH OUTPUT FROM READER
   8  RETRIEVE  - RETRIEVE SPF/TSO LIBRARIES FROM TAPE
   9  SCRIPT    - FORMAT, DISPLAY, AND OPTIONALLY PRINT SCRIPT TEXT
```

Figure 47. Utility selection menu

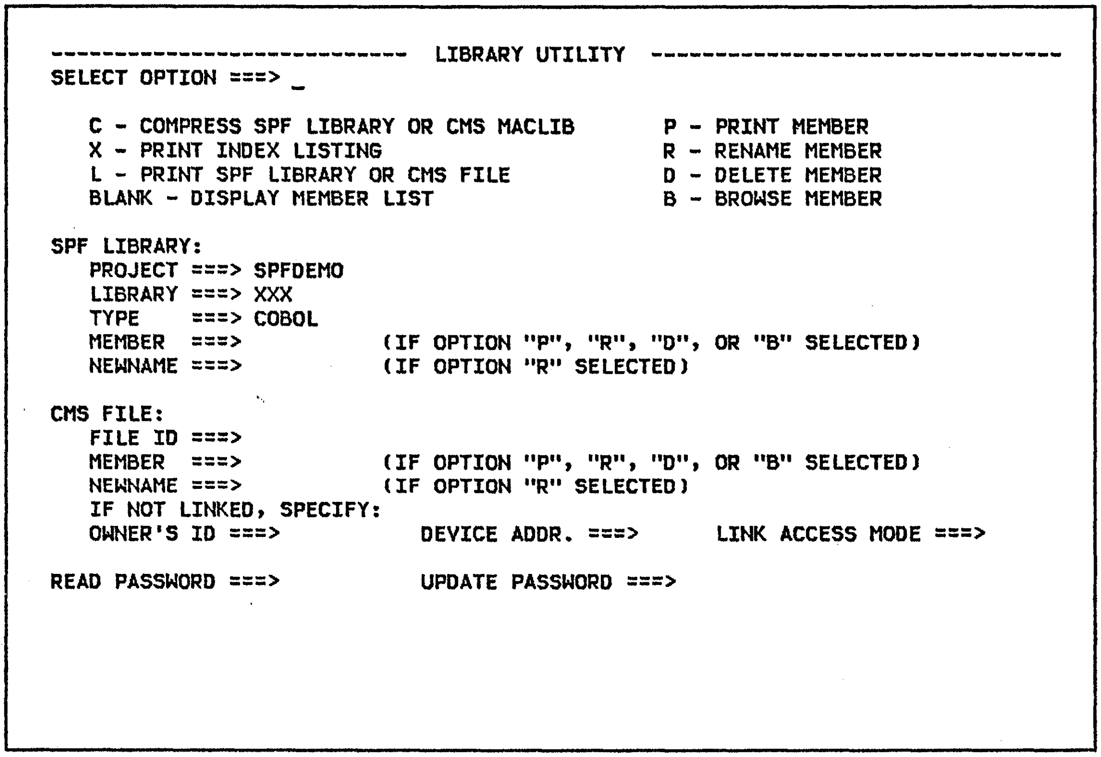
The following describes each of the utility functions, corresponding to each option on the secondary menu:

LIBRARY UTILITY (OPTION 3.1)

When this option is selected (utility option 1), a menu is displayed which allows the user to specify a file and indicate an action to be performed (Figure 48). Possible actions are:

C	- Compress SPF library or CMS MACLIB
X	- Print index listing
L	- Print SPF library or CMS file
P	- Print member
R	- Rename member
D	- Delete member
B	- Browse member
blank	- Display member list

Note: The library utility is intended primarily for maintenance of SPF libraries, MACLIBs, and TXTLIBs, but the "print index listing" and "print SPF library or CMS file" functions also apply to sequential CMS files.



```
---------------------------- LIBRARY UTILITY --------------------------------
SELECT OPTION ===> _

   C - COMPRESS SPF LIBRARY OR CMS MACLIB         P - PRINT MEMBER
   X - PRINT INDEX LISTING                        R - RENAME MEMBER
   L - PRINT SPF LIBRARY OR CMS FILE              D - DELETE MEMBER
   BLANK - DISPLAY MEMBER LIST                    B - BROWSE MEMBER

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX
   TYPE    ===> COBOL
   MEMBER  ===>          (IF OPTION "P", "R", "D", OR "B" SELECTED)
   NEWNAME ===>          (IF OPTION "R" SELECTED)

CMS FILE:
   FILE ID ===>
   MEMBER  ===>          (IF OPTION "P", "R", "D", OR "B" SELECTED)
   NEWNAME ===>          (IF OPTION "R" SELECTED)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>          UPDATE PASSWORD ===>
```

Figure 48. Library utility menu

If option "C" (compress SPF library or CMS MACLIB) is selected, any MACLIB (including an SPF library with a MACLIB organization) may be specified. The compress is accomplished by invoking the MACLIB command.

If option "X" (print index listing) is selected, any SPF library or CMS file may be specified. The index listing is recorded in the SPF list file. For SPF libraries, MACLIB's, and TXTLIBs, the index listing includes general information about the library followed by a member list. For a sequential CMS file, the index listing includes general information only. See Appendix C for an example of index listing format.

If option "L" (print SPF library or CMS file) is selected, any SPF library or CMS file may be specified. A source listing of the entire file (including all members), preceded by an index listing, is recorded in the SPF list file.

If option "P" (print member) is selected, a member of any SPF library, MACLIB, or TXTLIB may be specified. A source listing of the member is recorded in the SPF list file.

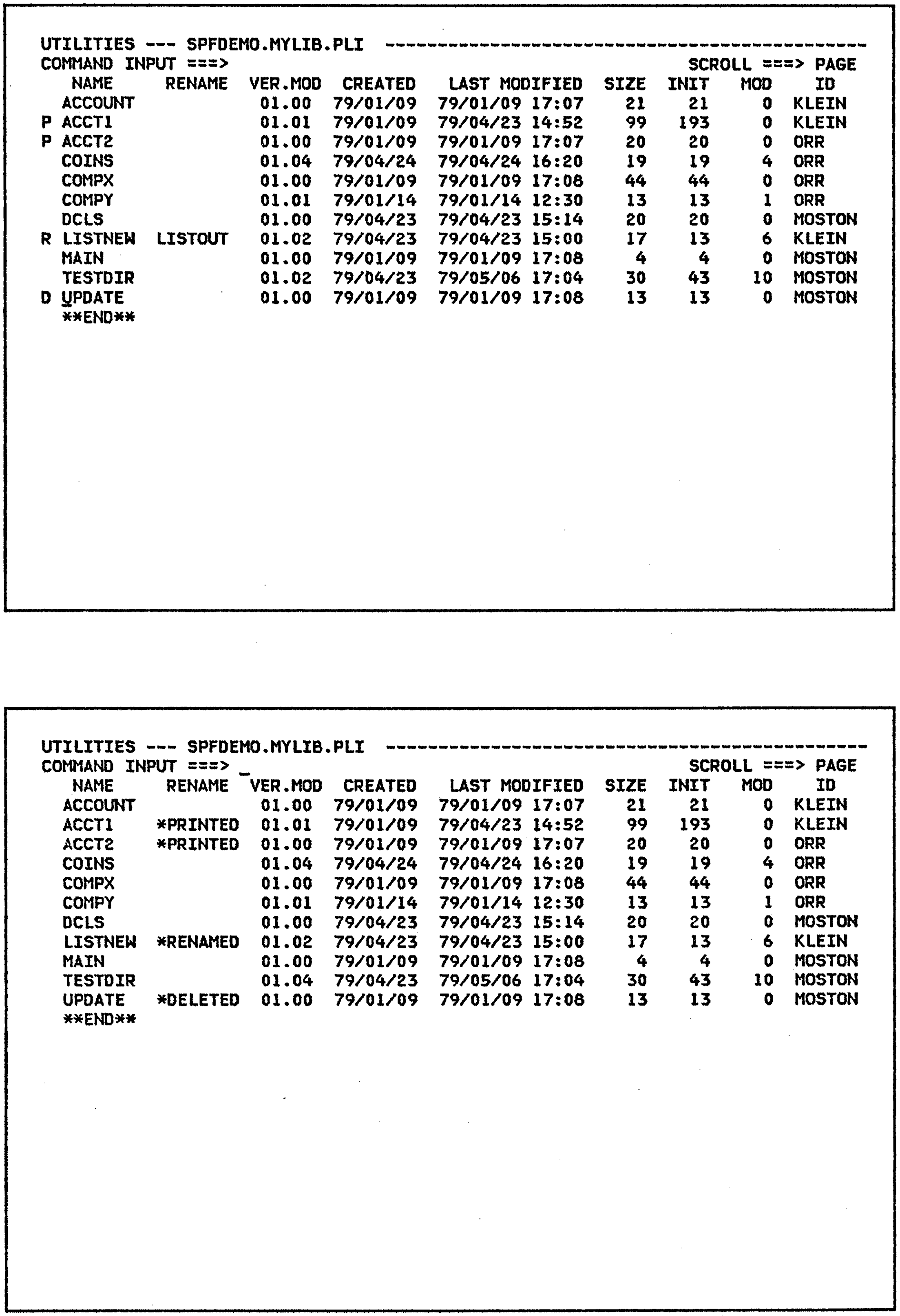
If option "R" or option "D" (rename member or delete member) is selected, a member of any SPF library, MACLIB, or TXTLIB may be specified. A new member name must also be specified for the "rename member" function.

If option "B" (browse member) is selected, a member of any SPF library, MACLIB, or TXTLIB may be specified. The specified member will be displayed in browse mode. All browse commands can be executed. When browse is terminated by pressing the End PF key, the menu from which the member was selected will be redisplayed.

If no option is specified (blank to display member listing), any SPF library, MACLIB, or TXTLIB may be specified. A member listing is then displayed from which the user may select members for printing, renaming, deleting, or browsing by entering "P", "R", "D", or "B" respectively in front of one or more member names. For renaming, a new member name must also be entered in the field immediately following the current member name.

The member list may be scrolled up and down via the scroll PF keys or via the LOCATE command. The member list may be terminated by pressing the End PF key, which causes a return to the previous menu.

Figure 49 shows a before-and-after example where members ACCT1 and ACCT2 are printed, member UPDATE is deleted, and member LISTNEW is renamed LISTOUT.



```
UTILITIES --- SPFDEMO.MYLIB.PLI ---------------------------------------------
COMMAND INPUT ===>                                            SCROLL ===> PAGE
   NAME     RENAME   VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD    ID
  ACCOUNT              01.00  79/01/09  79/01/09 17:07   21    21     0  KLEIN
P ACCT1                01.01  79/01/09  79/04/23 14:52   99   193     0  KLEIN
P ACCT2                01.00  79/01/09  79/01/09 17:07   20    20     0  ORR
  COINS                01.04  79/04/24  79/04/24 16:20   19    19     4  ORR
  COMPX                01.00  79/01/09  79/01/09 17:08   44    44     0  ORR
  COMPY                01.01  79/01/14  79/01/14 12:30   13    13     1  ORR
  DCLS                 01.00  79/04/23  79/04/23 15:14   20    20     0  MOSTON
R LISTNEW  LISTOUT     01.02  79/04/23  79/04/23 15:00   17    13     6  KLEIN
  MAIN                 01.00  79/01/09  79/01/09 17:08    4     4     0  MOSTON
  TESTDIR              01.02  79/04/23  79/05/06 17:04   30    43    10  MOSTON
D UPDATE               01.00  79/01/09  79/01/09 17:08   13    13     0  MOSTON
  **END**
```

```
UTILITIES --- SPFDEMO.MYLIB.PLI ---------------------------------------------
COMMAND INPUT ===> _                                          SCROLL ===> PAGE
   NAME     RENAME   VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD    ID
  ACCOUNT              01.00  79/01/09  79/01/09 17:07   21    21     0  KLEIN
  ACCT1    *PRINTED    01.01  79/01/09  79/04/23 14:52   99   193     0  KLEIN
  ACCT2    *PRINTED    01.00  79/01/09  79/01/09 17:07   20    20     0  ORR
  COINS                01.04  79/04/24  79/04/24 16:20   19    19     4  ORR
  COMPX                01.00  79/01/09  79/01/09 17:08   44    44     0  ORR
  COMPY                01.01  79/01/14  79/01/14 12:30   13    13     1  ORR
  DCLS                 01.00  79/04/23  79/04/23 15:14   20    20     0  MOSTON
  LISTNEW  *RENAMED    01.02  79/04/23  79/04/23 15:00   17    13     6  KLEIN
  MAIN                 01.00  79/01/09  79/01/09 17:08    4     4     0  MOSTON
  TESTDIR              01.04  79/04/23  79/05/06 17:04   30    43    10  MOSTON
  UPDATE   *DELETED    01.00  79/01/09  79/01/09 17:08   13    13     0  MOSTON
  **END**
```

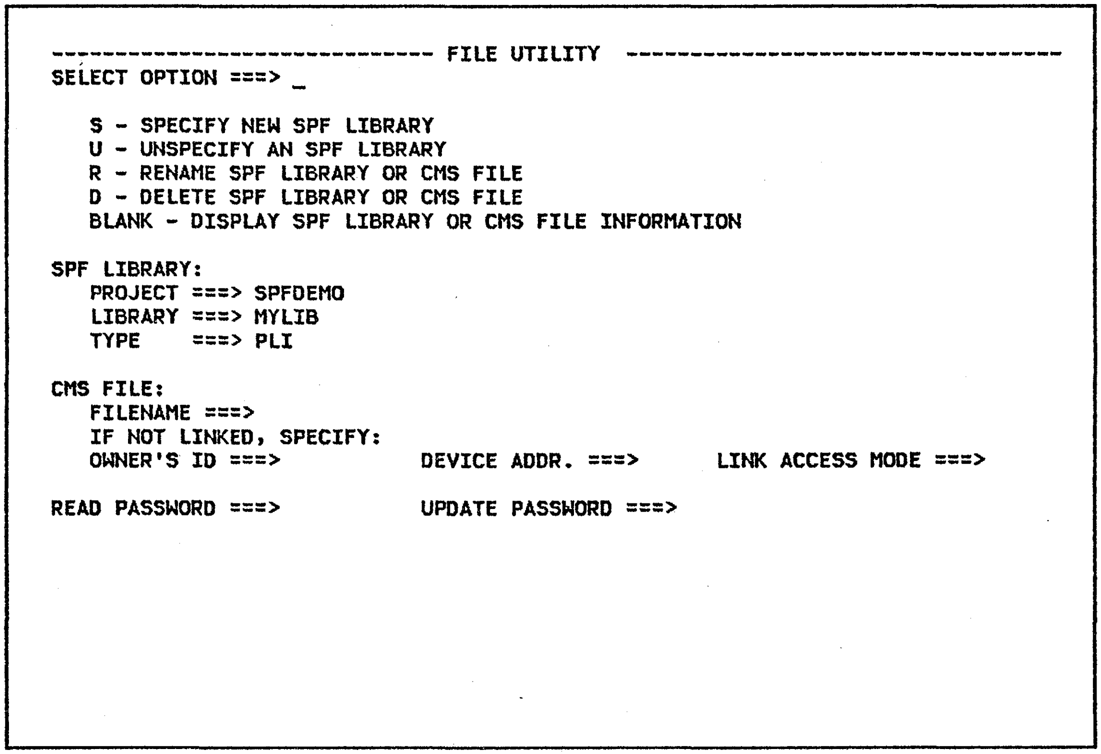
Figure 49. Library utility - print, rename, and delete

FILE UTILITY (OPTION 3.2)

When this option is selected (utility option 2), a menu is displayed which allows the user to specify a file and indicate an action to be performed (Figure 50). Possible actions are:

```
S     - Specify new SPF library
U     - Unspecify an SPF library
R     - Rename file or SPF library
D     - Delete ("erase") file or SPF library
blank - Display file or SPF library information
```

For options "S" and "U", an SPF library identification (project, library, and type) must be specified. For the other options, any SPF library or CMS file may be specified.



```
------------------------------ FILE UTILITY ----------------------------------
SELECT OPTION ===> _

   S - SPECIFY NEW SPF LIBRARY
   U - UNSPECIFY AN SPF LIBRARY
   R - RENAME SPF LIBRARY OR CMS FILE
   D - DELETE SPF LIBRARY OR CMS FILE
   BLANK - DISPLAY SPF LIBRARY OR CMS FILE INFORMATION

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB
   TYPE    ===> PLI

CMS FILE:
   FILENAME ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>          UPDATE PASSWORD ===>
```

Figure 50. File utility menu

If option "S" (specify new SPF library) is selected, a menu is displayed to allow specification of SPF library attributes and location (minidisk), link access mode for update, and corresponding CMS filetype/filename (Figure 51). Default values are pre-entered based on what the user last entered, or based on the last "display file information" request (whichever occurred most recently). The user may overtype the displayed defaults.

See Appendix B for guidelines in specifying SPF libraries.

If option "U" (unspecify SPF library) is selected, a confirmation menu is displayed to ensure that the user did not select this option by mistake. As directed in the menu, the user must then press either the ENTER key to confirm or the End PF key to cancel. Either action causes a return to the previous menu.

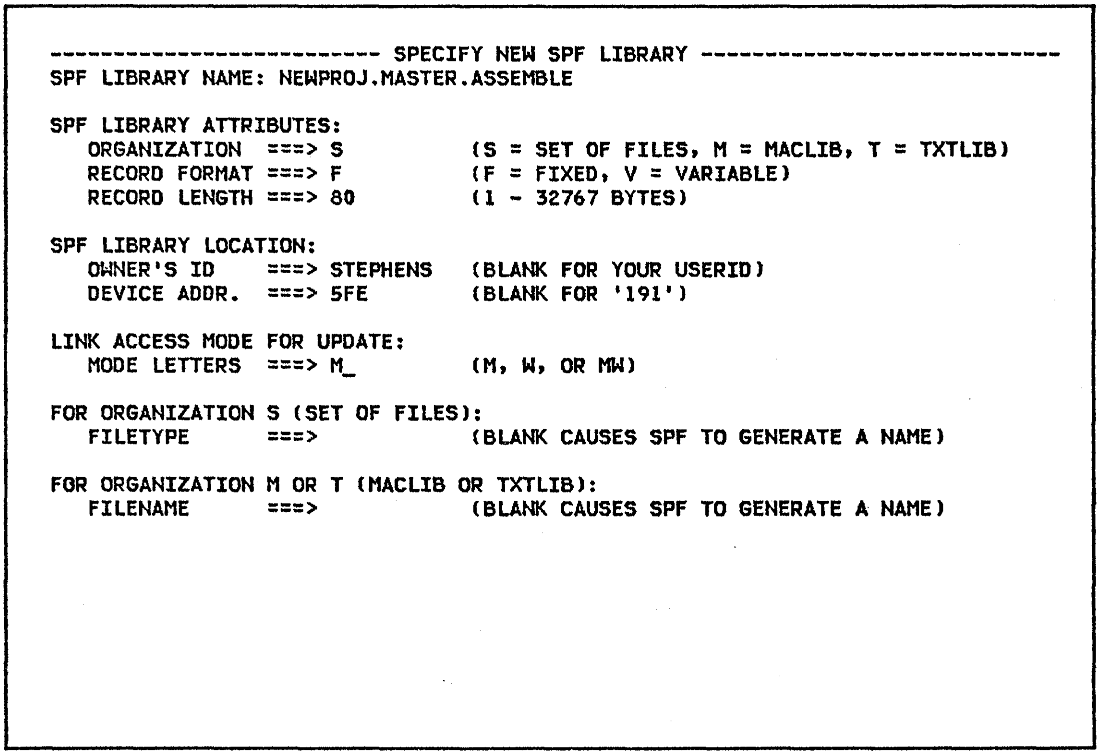
Option "U" causes the information about the SPF library to be purged from the SPF catalog, but it does not cause the associated file(s) to be deleted.

If option "R" (rename file or SPF library) is selected, a menu is displayed to allow the user to enter the new file or SPF library identification (Figure 52). The user may type the new identifier(s) and press ENTER (to rename), or press the End PF key to cancel. Either action causes a return to the previous menu.

If option "D" (delete file or SPF library) is selected, a confirmation menu is displayed to ensure that the user did not select this option by mistake. As directed in the menu, the user must then press either the ENTER key to confirm or the End PF key to cancel. Either action causes a return to the previous menu.

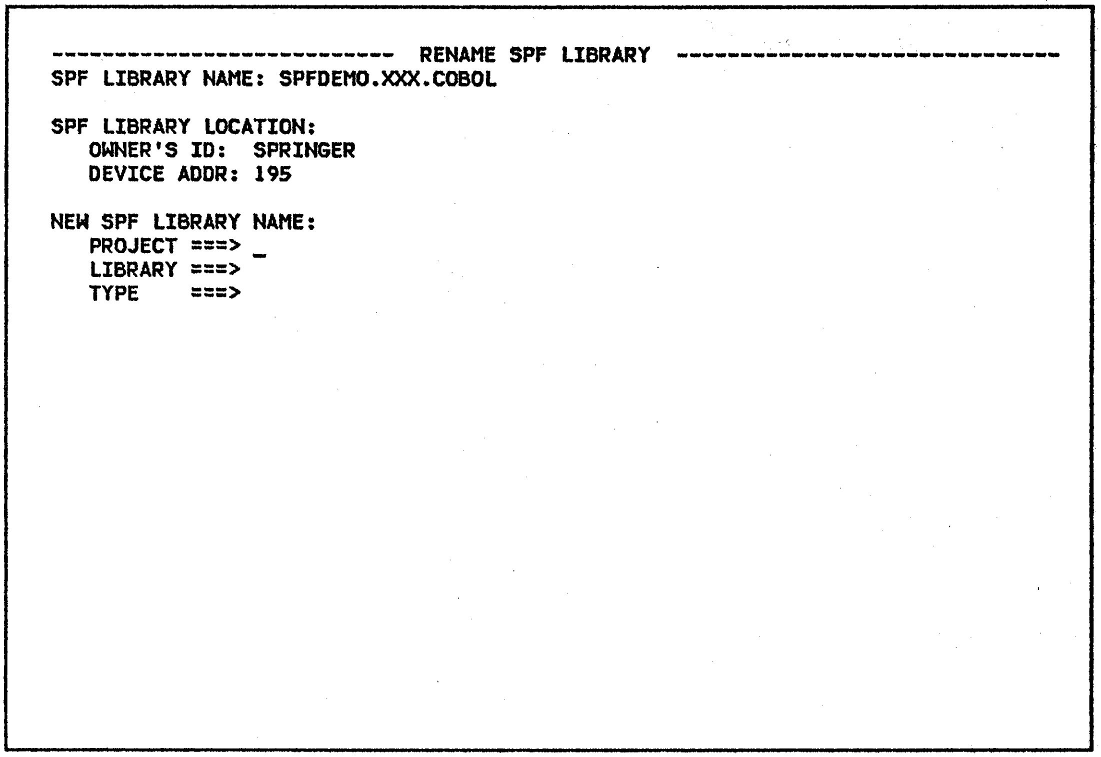
If an SPF library is specified, option "D" causes the library to be unspecified and the associated file(s) to be deleted (erased).

If no option is specified (blank to display file or SPF library information), the location, characteristics, and current space utilization of the specified library or file are displayed (Figure 53). The user may return to the previous menu by pressing either the ENTER or End PF key.



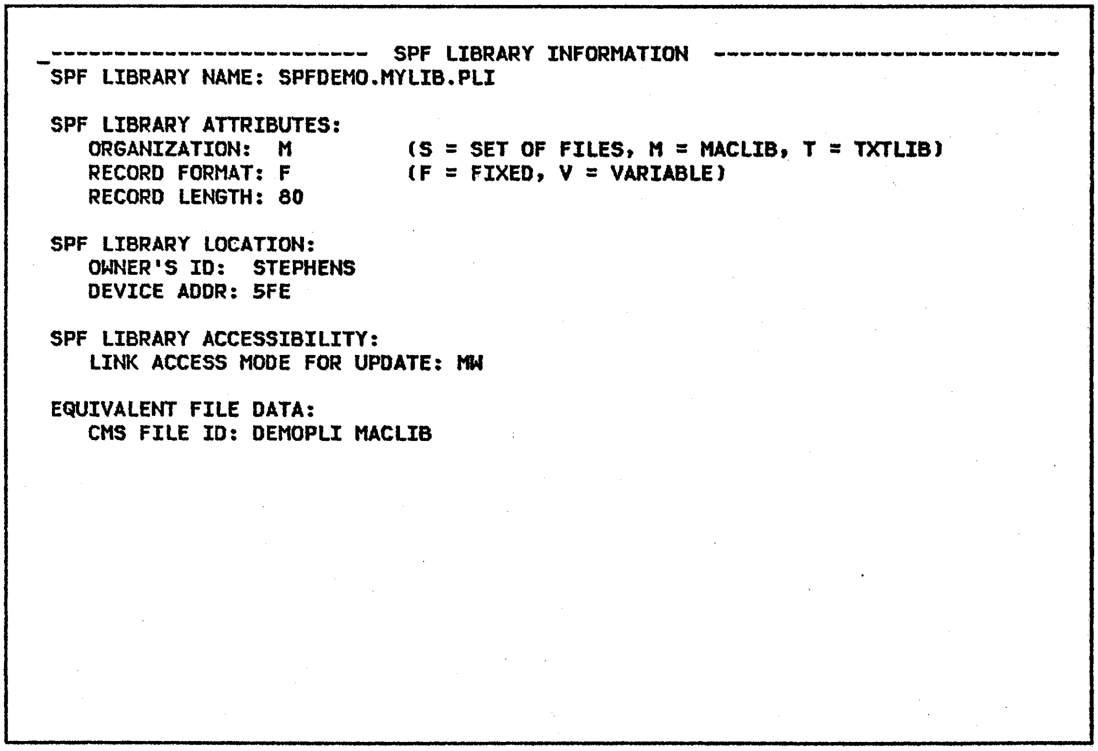
```
-------------------------- SPECIFY NEW SPF LIBRARY ---------------------------
SPF LIBRARY NAME: NEWPROJ.MASTER.ASSEMBLE

SPF LIBRARY ATTRIBUTES:
   ORGANIZATION  ===> S          (S = SET OF FILES, M = MACLIB, T = TXTLIB)
   RECORD FORMAT ===> F          (F = FIXED, V = VARIABLE)
   RECORD LENGTH ===> 80         (1 - 32767 BYTES)

SPF LIBRARY LOCATION:
   OWNER'S ID    ===> STEPHENS   (BLANK FOR YOUR USERID)
   DEVICE ADDR.  ===> 5FE        (BLANK FOR '191')

LINK ACCESS MODE FOR UPDATE:
   MODE LETTERS  ===> M_         (M, W, OR MW)

FOR ORGANIZATION S (SET OF FILES):
   FILETYPE      ===>            (BLANK CAUSES SPF TO GENERATE A NAME)

FOR ORGANIZATION M OR T (MACLIB OR TXTLIB):
   FILENAME      ===>            (BLANK CAUSES SPF TO GENERATE A NAME)
```

Figure 51. File utility - specify new SPF library



```
---------------------------- RENAME SPF LIBRARY ------------------------------
SPF LIBRARY NAME: SPFDEMO.XXX.COBOL

SPF LIBRARY LOCATION:
   OWNER'S ID:  SPRINGER
   DEVICE ADDR: 195

NEW SPF LIBRARY NAME:
   PROJECT ===> _
   LIBRARY ===>
   TYPE    ===>
```

Figure 52. File utility - rename SPF library



```
_------------------------- SPF LIBRARY INFORMATION ---------------------------
 SPF LIBRARY NAME: SPFDEMO.MYLIB.PLI

 SPF LIBRARY ATTRIBUTES:
    ORGANIZATION:  M        (S = SET OF FILES, M = MACLIB, T = TXTLIB)
    RECORD FORMAT: F        (F = FIXED, V = VARIABLE)
    RECORD LENGTH: 80

 SPF LIBRARY LOCATION:
    OWNER'S ID:  STEPHENS
    DEVICE ADDR: 5FE

 SPF LIBRARY ACCESSIBILITY:
    LINK ACCESS MODE FOR UPDATE: MW

 EQUIVALENT FILE DATA:
    CMS FILE ID: DEMOPLI MACLIB
```

Figure 53. File utility - display information

MOVE/COPY UTILITY (OPTION 3.3)

When this option is selected (utility option 3), a menu is displayed which allows the user to specify the "from" file (and member if it is an SPF library, MACLIB, or TXTLIB), and to indicate an action to be performed (Figure 54 top). Possible actions are:

CP	- Copy file or member(s) and print
MP	- Move file or member(s) and print
C	- Copy without printing
M	- Move without printing

When ENTER is pressed, a second menu is displayed which allows the user to specify the "to" file (Figure 54 bottom). If the "to" file is an SPF library, MACLIB, or TXTLIB, the option to replace like-named members (YES or NO) must be specified. If the "to" file is sequential, a disposition (OLD or MOD) must be specified. A disposition of OLD causes an existing file to be rewritten. A disposition of MOD causes the "from" data to be appended to the end of an existing "to" file.

Any SPF library or CMS file may be specified for either the "from" or "to" file, but the same SPF library, MACLIB, or TXTLIB may not be specified as both the "from" and "to" file.

If the "to" file is a CMS file which does not currently exist, a new file will be created with the same attributes as the "from" file. If the "to" file is an SPF library, it must have been previously specified via option 3.2 (specify SPF library).

If "move" is indicated and the "from" file is an SPF library, MACLIB, or TXTLIB, the selected member(s) will be deleted in the "from" file after they have been successfully copied to the "to" file. If "move" is indicated and the "from" file is a sequential CMS file, the entire "from" file will be deleted after its contents have been successfully copied to the "to" file. If "copy" is indicated, no deletions will occur.

If "print" is indicated for either move or copy, source listing(s) will be recorded in the SPF list file. If the "to" file is an SPF library, MACLIB, or TXTLIB, a listing of each new or replaced member will be recorded. If the "to" file is a sequential CMS file, a listing of its entire contents will be recorded after the information has been successfully copied.

The following rules apply for specifying member names if the "from" file is an SPF library, MACLIB, or TXTLIB:

- Specifying a valid "from" member name causes a single member to be moved or copied.

- Leaving the "from" member name blank causes a member listing to be displayed (after the second menu has been displayed).

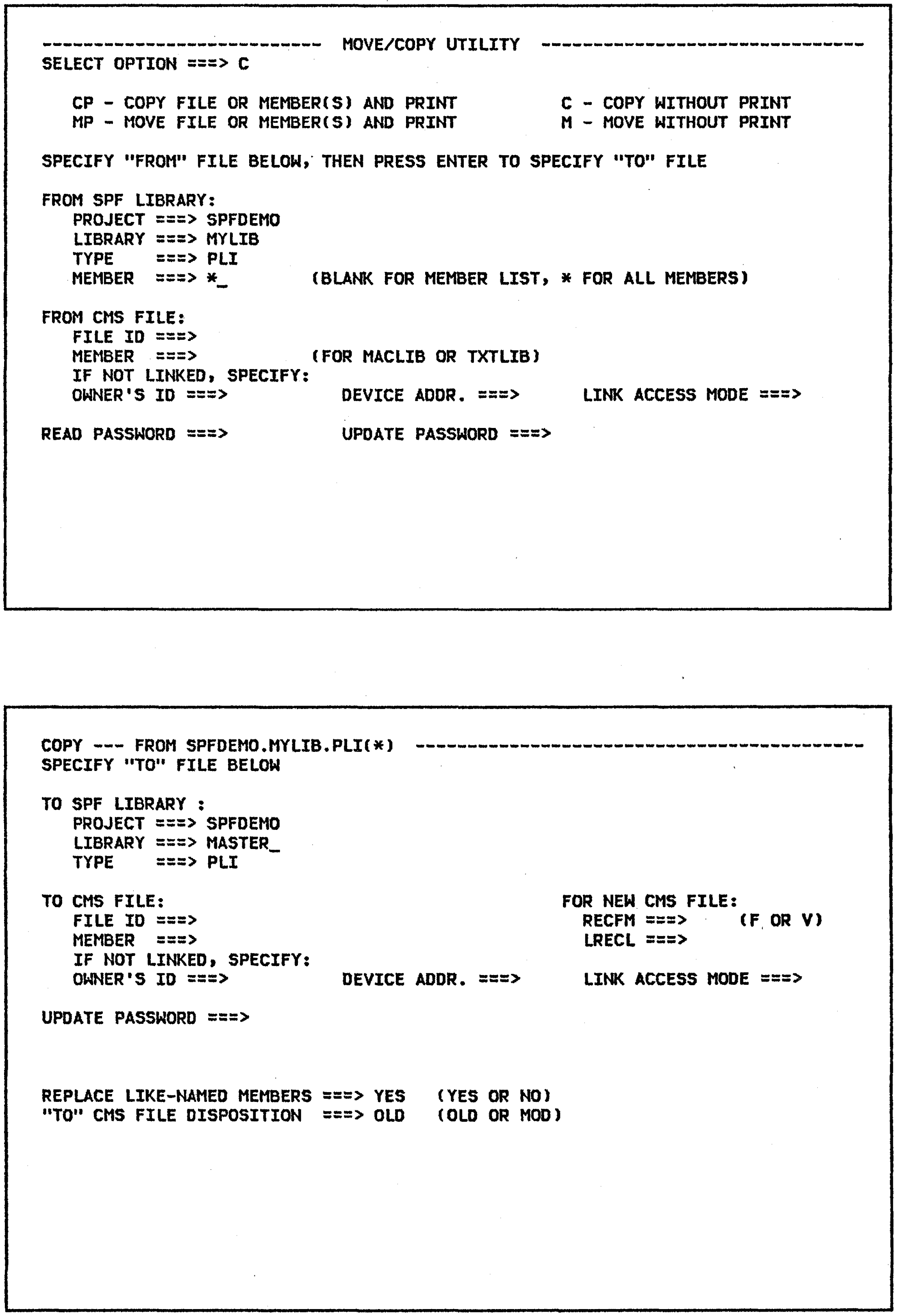
- Specifying a "from" member name of asterisk (*) causes all members to be moved or copied.

If the "to" file is an SPF library, MACLIB, or TXTLIB, the "to" member name must be specified for the following conditions:

- The "from" file is a sequential CMS file.

- A single member is to be moved or copied and the member is to be renamed in the "to" file. (Members may also be renamed from the member listing, see below.)

In all other cases, the "to" member should be left blank.



```
--------------------------- MOVE/COPY UTILITY -------------------------------
SELECT OPTION ===> C

   CP - COPY FILE OR MEMBER(S) AND PRINT          C - COPY WITHOUT PRINT
   MP - MOVE FILE OR MEMBER(S) AND PRINT          M - MOVE WITHOUT PRINT

SPECIFY "FROM" FILE BELOW, THEN PRESS ENTER TO SPECIFY "TO" FILE

FROM SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB
   TYPE    ===> PLI
   MEMBER  ===> *_        (BLANK FOR MEMBER LIST, * FOR ALL MEMBERS)

FROM CMS FILE:
   FILE ID ===>
   MEMBER  ===>           (FOR MACLIB OR TXTLIB)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>           UPDATE PASSWORD ===>
```

```
COPY --- FROM SPFDEMO.MYLIB.PLI(*) -------------------------------------------
SPECIFY "TO" FILE BELOW

TO SPF LIBRARY :
   PROJECT ===> SPFDEMO
   LIBRARY ===> MASTER_
   TYPE    ===> PLI

TO CMS FILE:                                        FOR NEW CMS FILE:
   FILE ID ===>                                       RECFM ===>     (F OR V)
   MEMBER  ===>                                       LRECL ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

UPDATE PASSWORD ===>



REPLACE LIKE-NAMED MEMBERS ===> YES   (YES OR NO)
"TO" CMS FILE DISPOSITION  ===> OLD   (OLD OR MOD)
```

Figure 54. Move/Copy utility menus

A member listing is displayed when the "from" file is an SPF library, MACLIB, or TXTLIB, and the "from" member name is left blank. The user may select members to be moved or copied by entering an "S" in front of one or more member names. The options for move or copy, and print or no print must have been previously specified on the first move/copy file menu. If the "to" file is an SPF library, MACLIB, or TXTLIB, the member(s) may also be renamed by entering a new member name in the field immediately following the current name.

The member list may be scrolled up and down via the Scroll PF keys or via the LOCATE command. The member list may be terminated by pressing the End PF key, which causes a return to the first move/copy menu.

Figure 55 shows a before-and-after example where members ACCOUNT, ACCT1, ACCT2, and COMPY are copied to another library, and member COMPY is also renamed ZCOMP.

For any move/copy operation, the following rules apply.

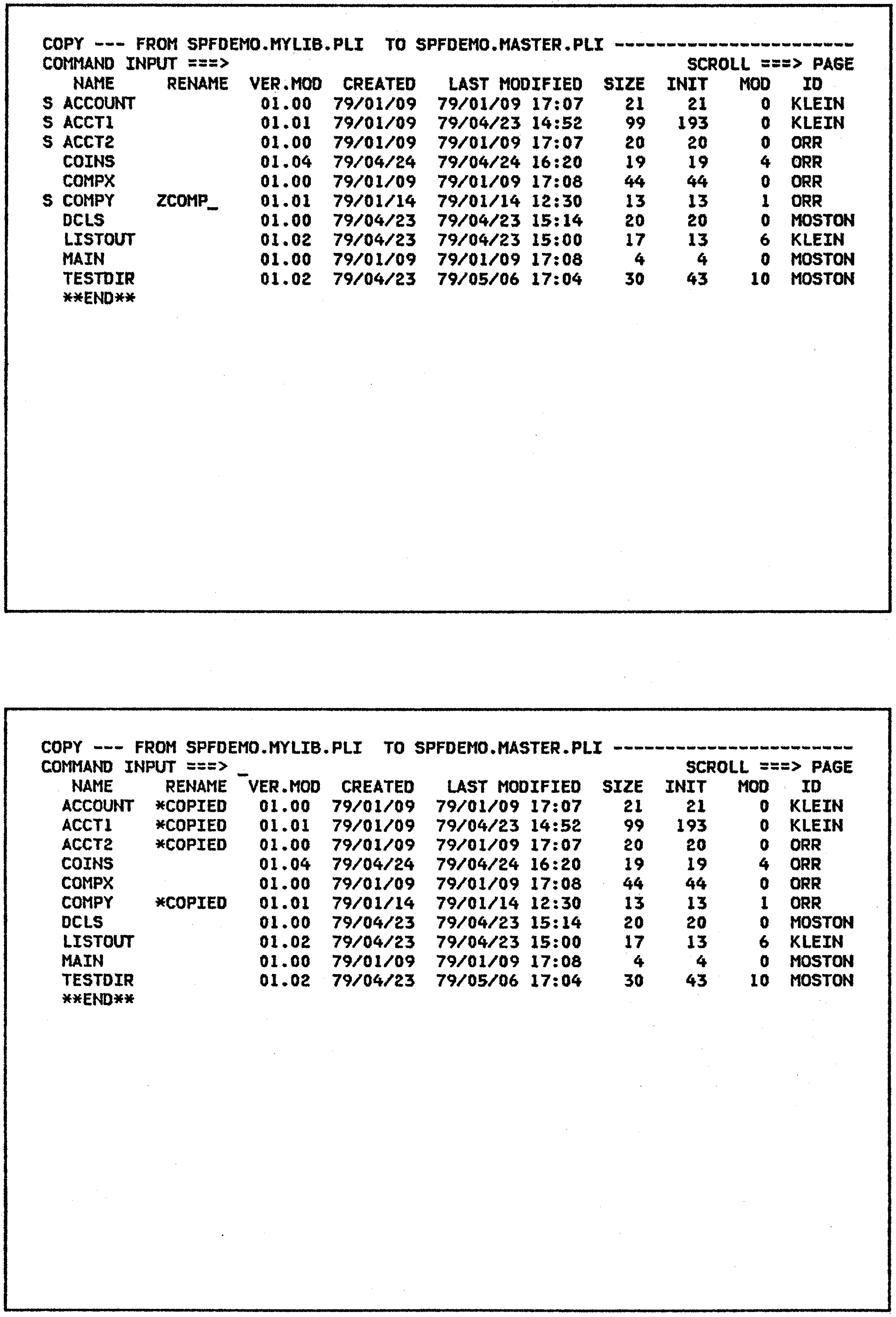
- If the "to" file is sequential, its entire contents are replaced (if disposition OLD was specified) or the new information is added to the end (if disposition MOD was specified). If multiple members of an SPF library, MACLIB, or TXTLIB are moved or copied to a sequential CMS file by using a member selection list or by specifying all members (*), they are merged in the "to" file regardless of the disposition.

- If the "to" file is an SPF library, MACLIB, or TXTLIB, new members are added except when the member names are alike; like-named members are replaced (if replace YES was specified) or not copied (if replace NO was specified).

- Record formats and logical record lengths for the two files need not be alike. When necessary, data will be truncated or right-padded with blanks to accommodate different record lengths.

- The data to be moved or copied is not renumbered or modified in any way, except for possible truncation or padding as noted above. Printer control characters, if present, are treated as part of the data.

- SPF library statistics are not modified when moving or copying between SPF libraries.



```
COPY --- FROM SPFDEMO.MYLIB.PLI  TO SPFDEMO.MASTER.PLI -----------------------
COMMAND INPUT ===>                                             SCROLL ===> PAGE
   NAME      RENAME   VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
S ACCOUNT              01.00  79/01/09  79/01/09 17:07    21    21     0  KLEIN
S ACCT1                01.01  79/01/09  79/04/23 14:52    99   193     0  KLEIN
S ACCT2                01.00  79/01/09  79/01/09 17:07    20    20     0  ORR
  COINS                01.04  79/04/24  79/04/24 16:20    19    19     4  ORR
  COMPX                01.00  79/01/09  79/01/09 17:08    44    44     0  ORR
S COMPY     ZCOMP_     01.01  79/01/14  79/01/14 12:30    13    13     1  ORR
  DCLS                 01.00  79/04/23  79/04/23 15:14    20    20     0  MOSTON
  LISTOUT              01.02  79/04/23  79/04/23 15:00    17    13     6  KLEIN
  MAIN                 01.00  79/01/09  79/01/09 17:08     4     4     0  MOSTON
  TESTDIR              01.02  79/04/23  79/05/06 17:04    30    43    10  MOSTON
  **END**
```

```
COPY --- FROM SPFDEMO.MYLIB.PLI  TO SPFDEMO.MASTER.PLI -----------------------
COMMAND INPUT ===> _                                           SCROLL ===> PAGE
   NAME      RENAME   VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT   *COPIED    01.00  79/01/09  79/01/09 17:07    21    21     0  KLEIN
  ACCT1     *COPIED    01.01  79/01/09  79/04/23 14:52    99   193     0  KLEIN
  ACCT2     *COPIED    01.00  79/01/09  79/01/09 17:07    20    20     0  ORR
  COINS                01.04  79/04/24  79/04/24 16:20    19    19     4  ORR
  COMPX                01.00  79/01/09  79/01/09 17:08    44    44     0  ORR
  COMPY     *COPIED    01.01  79/01/14  79/01/14 12:30    13    13     1  ORR
  DCLS                 01.00  79/04/23  79/04/23 15:14    20    20     0  MOSTON
  LISTOUT              01.02  79/04/23  79/04/23 15:00    17    13     6  KLEIN
  MAIN                 01.00  79/01/09  79/01/09 17:08     4     4     0  MOSTON
  TESTDIR              01.02  79/04/23  79/05/06 17:04    30    43    10  MOSTON
  **END**
```

Figure 55. Move/Copy utility - copy members from list

PROJECT UTILITY (OPTION 3.4)

When this option is selected (utility option 4), a menu is displayed that allows the user to specify parameters for an SPF library and an action to be performed (Figure 56). Possible actions are:

```
P     - Print project entries
blank  - Display project entries
```

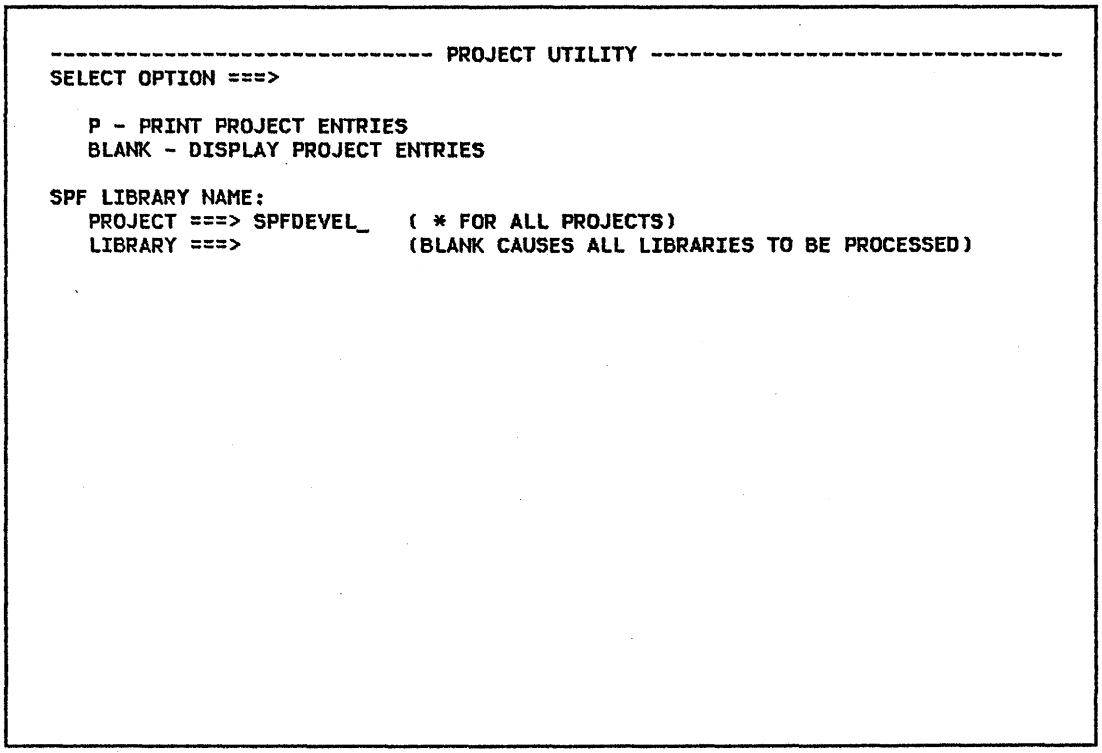
Information may be requested for:

- All SPF libraries known to the system, by specifying an asterisk (*) for the project name.
- All SPF libraries with a given project name, by specifying the project name and leaving the library name blank.
- All SPF libraries with a given project and library name.

If option "P" (print project entries) is selected, a project listing is recorded in the SPF list file. The listing contains CMS file identifications, locations, and file characteristics for each SPF library with the specified project name.

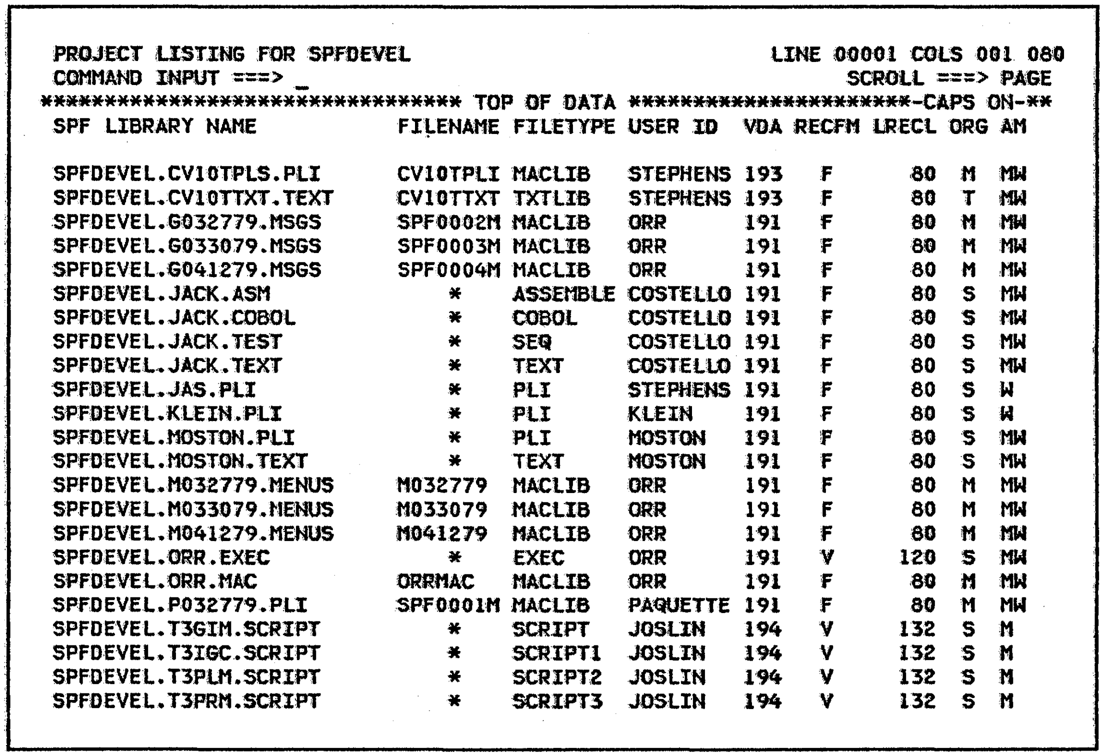
If no option is specified (blank to display project entries), a project listing is displayed at the terminal in browse mode. The listing may be scrolled up and down via the Scroll PF keys. All browse commands may be entered. The listing may be terminated by pressing the End PF key, causing a return to the previous menu.

Figure 57 shows an example of an SPF project listing.



```
------------------------------ PROJECT UTILITY -------------------------------
SELECT OPTION ===>

   P - PRINT PROJECT ENTRIES
   BLANK - DISPLAY PROJECT ENTRIES

SPF LIBRARY NAME:
   PROJECT ===> SPFDEVEL_    ( * FOR ALL PROJECTS)
   LIBRARY ===>              (BLANK CAUSES ALL LIBRARIES TO BE PROCESSED)
```

Figure 56. SPF project utility menu



```
 PROJECT LISTING FOR SPFDEVEL                            LINE 00001 COLS 001 080
 COMMAND INPUT ===> _                                          SCROLL ===> PAGE
********************************* TOP OF DATA *********************-CAPS ON-**
 SPF LIBRARY NAME           FILENAME FILETYPE USER ID  VDA RECFM LRECL ORG AM

 SPFDEVEL.CV10TPLS.PLI      CV10TPLI MACLIB   STEPHENS 193   F      80  M  MW
 SPFDEVEL.CV10TTXT.TEXT     CV10TTXT TXTLIB   STEPHENS 193   F      80  T  MW
 SPFDEVEL.G032779.MSGS      SPF0002M MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.G033079.MSGS      SPF0003M MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.G041279.MSGS      SPF0004M MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.JACK.ASM              *    ASSEMBLE COSTELLO 191   F      80  S  MW
 SPFDEVEL.JACK.COBOL            *    COBOL    COSTELLO 191   F      80  S  MW
 SPFDEVEL.JACK.TEST             *    SEQ      COSTELLO 191   F      80  S  MW
 SPFDEVEL.JACK.TEXT             *    TEXT     COSTELLO 191   F      80  S  MW
 SPFDEVEL.JAS.PLI               *    PLI      STEPHENS 191   F      80  S  W
 SPFDEVEL.KLEIN.PLI             *    PLI      KLEIN    191   F      80  S  W
 SPFDEVEL.MOSTON.PLI            *    PLI      MOSTON   191   F      80  S  MW
 SPFDEVEL.MOSTON.TEXT           *    TEXT     MOSTON   191   F      80  S  MW
 SPFDEVEL.M032779.MENUS     M032779  MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.M033079.MENUS     M033079  MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.M041279.MENUS     M041279  MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.ORR.EXEC              *    EXEC     ORR      191   V     120  S  MW
 SPFDEVEL.ORR.MAC           ORRMAC   MACLIB   ORR      191   F      80  M  MW
 SPFDEVEL.P032779.PLI       SPF0001M MACLIB   PAQUETTE 191   F      80  M  MW
 SPFDEVEL.T3GIM.SCRIPT          *    SCRIPT   JOSLIN   194   V     132  S  M
 SPFDEVEL.T3IGC.SCRIPT          *    SCRIPT1  JOSLIN   194   V     132  S  M
 SPFDEVEL.T3PLM.SCRIPT          *    SCRIPT2  JOSLIN   194   V     132  S  M
 SPFDEVEL.T3PRM.SCRIPT          *    SCRIPT3  JOSLIN   194   V     132  S  M
```

Figure 57. SPF project utility - sample listing

RESET SPF STATISTICS UTILITY (OPTION 3.5)

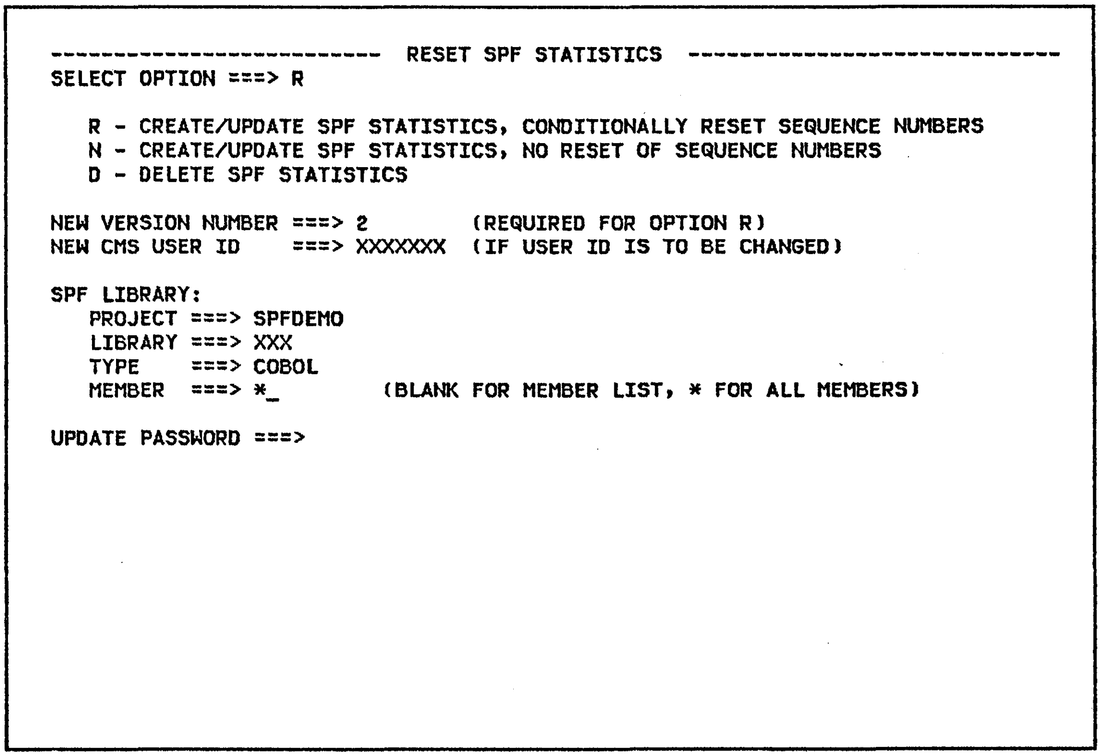
When this option is specified (utility option 5), a menu is displayed that allows the user to create, update, or delete SPF statistics and to reset sequence numbers (Figure 58). The reset utility handles SPF libraries only. Refer to the section entitled "Library Activity Statistics" on page 15 for a discussion of the type of information maintained for each member of an SPF library.

There are three valid options for this utility:

R - Create/update statistics, conditionally reset sequence numbers

N - Create/update statistics, no reset of sequence numbers

D - Delete SPF statistics



```
-------------------------- RESET SPF STATISTICS ----------------------------
SELECT OPTION ===> R

   R - CREATE/UPDATE SPF STATISTICS, CONDITIONALLY RESET SEQUENCE NUMBERS
   N - CREATE/UPDATE SPF STATISTICS, NO RESET OF SEQUENCE NUMBERS
   D - DELETE SPF STATISTICS

NEW VERSION NUMBER ===> 2        (REQUIRED FOR OPTION R)
NEW CMS USER ID    ===> XXXXXXX  (IF USER ID IS TO BE CHANGED)

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX
   TYPE    ===> COBOL
   MEMBER  ===> *_        (BLANK FOR MEMBER LIST, * FOR ALL MEMBERS)

UPDATE PASSWORD ===>
```

Figure 58. Reset utility menu

The "R" option is used to either create SPF statistics in a library which does not currently have them, or to update SPF statistics in a library. The data is scanned to determine if valid, ascending sequence numbers are present in all records. If so, the data is renumbered and the modification level flags (the last two digits of each sequence number) are set to zeros. If valid sequence numbers are not present, renumbering is not done.

The "N" option is also used to create or update SPF statistics, as in option "R"; however the data is not renumbered. This option should be used if the data already contains SPF statistics and the user wants to update the user id and/or version information without renumbering the data.

The "D" option is used to delete SPF statistics.

The version number field is required for option "R", optional for option "N", and ignored for option "D". If option "N" is selected and a member without SPF statistics is selected, the statistics are created as if the version had been specified as "1". If a version number is specified, the statistics are created or reset as follows:

Version number:	Set to the specified value
Modification level:	Set to zero
Creation date:	Set to current date
Date/time last modified:	Set to current date and time
Current number of lines:	Set to the current number of lines
Initial number lines:	Set to the current number of lines
Number of modified lines:	Set to zero

The user id field is optional for options "R" or "N"; it is ignored for option "D". If a user id is specified, it will be placed in the user id field of the SPF statistics. If the user id is left blank and a member without SPF statistics is selected, the user id field of the statistics will be set to the current user id.

The rules for specifying member names are as follows:

- Specifying a valid member name causes the statistics to be created or updated for a single member.

- Leaving the member name blank causes a member listing to be displayed.

- Specifying a member name of asterisk (*) causes the statistics to be created or updated for all members in the library.

If member listing is requested, the user may select members to be reset by entering an "S" in front of one or more member names. The member list may be scrolled up and down via the Scroll PF keys or via the LOCATE command. The member list may be terminated by pressing the End PF key, which causes a return to the previous menu.

SPOOL UTILITY (OPTION 3.6)

When this option is selected (utility option 6), a menu is displayed which allows the user to specify an SPF library or CMS file and indicate an action to be performed (Figure 59).

<u>Note</u>: The SPF list and log files may not be specified; they may only be printed upon SPF termination.

Possible actions are:

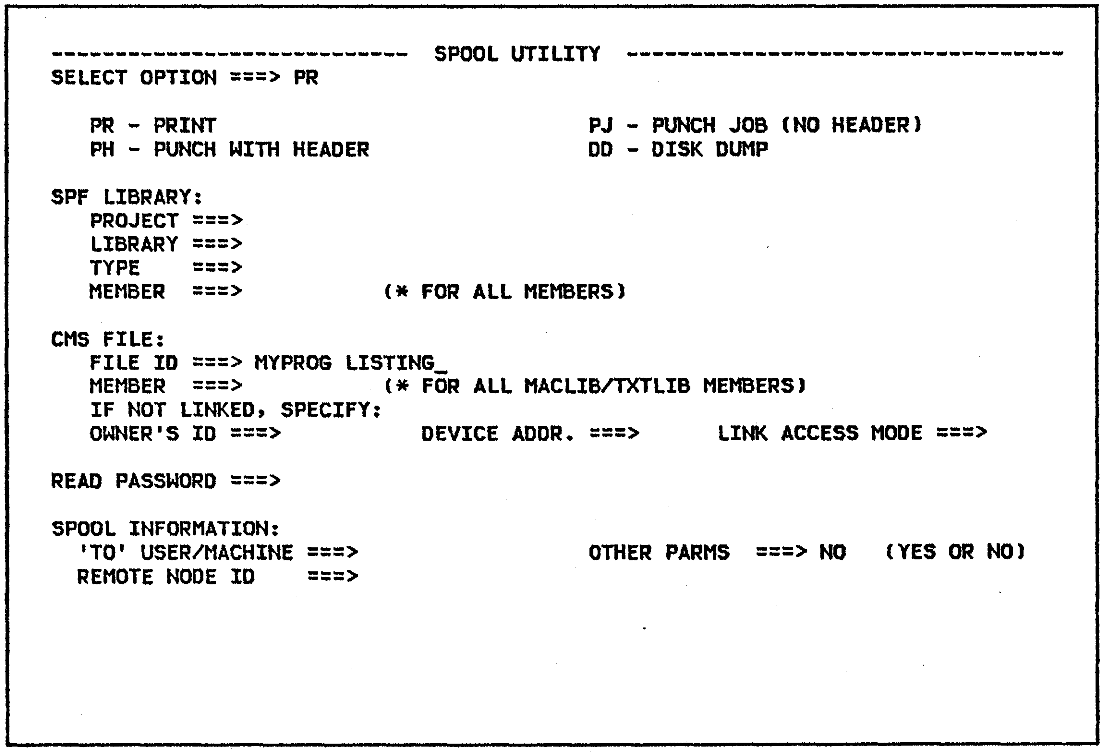
PR	- Print
PH	- Punch with header
PJ	- Punch job (no` header)
DD	- Disk dump

Option "PR" is used to print a listing file. Normally, option "DD" is used to send a file to another user. Option "PH" may be used instead of "DD" if the file has a record length of 80 or less. Option "PJ" is used to send a file to another machine for job execution (may be used for sending CMS jobs to a CMS batch machine, or JCL to a VS or DOS machine).

An asterisk (*) may be entered in the member name field to process all members of an SPF library or CMS MACLIB or TXTLIB.

The menu also allows the user to specify:

- To user/machine - if the data is to be spooled to the virtual reader of another user or machine.
- Remote node id - if the other virtual machine is at a remote location on the network.
- Other parms (yes/no) - to allow specification of additional parameters and options.



```
----------------------------  SPOOL UTILITY  ----------------------------------
SELECT OPTION ===> PR

   PR - PRINT                              PJ - PUNCH JOB (NO HEADER)
   PH - PUNCH WITH HEADER                  DD - DISK DUMP

SPF LIBRARY:
   PROJECT ===>
   LIBRARY ===>
   TYPE    ===>
   MEMBER  ===>          (* FOR ALL MEMBERS)

CMS FILE:
   FILE ID ===> MYPROG LISTING_
   MEMBER  ===>          (* FOR ALL MACLIB/TXTLIB MEMBERS)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>

SPOOL INFORMATION:
  'TO' USER/MACHINE ===>                   OTHER PARMS  ===> NO   (YES OR NO)
  REMOTE NODE ID    ===>
```

Figure 59. Spool utility - first menu

When the user enters the desired information, a second menu will be displayed (Figure 60) if "other parms - yes" was specified on the first menu. Information on the second menu is remembered and pre-filled from what was entered the last time, except for "to user/machine" and "remote node id", which are simply copied from the first menu.

Information which may be entered on the second menu is:

- Print command options - if option "PR" was specified. See description of PRINT command in the VM/370 CMS Command Reference, GC20-1818.

- Number of copies - if more than one copy is desired.

- Spool class - for other than class A.

- For user - to spool the output to another user's virtual printer on the same VM system (ignored if either "to user/machine" or "remote node id" is specified).

- 3800 keywords - if output is directed to an IBM 3800 printer which is attached to the CMS system as a virtual spooling device. See description of SPOOL command in the VM/370 CP Command Reference, GC20-1820.

- To user/machine - to spool the output to the virtual reader of another user or machine.

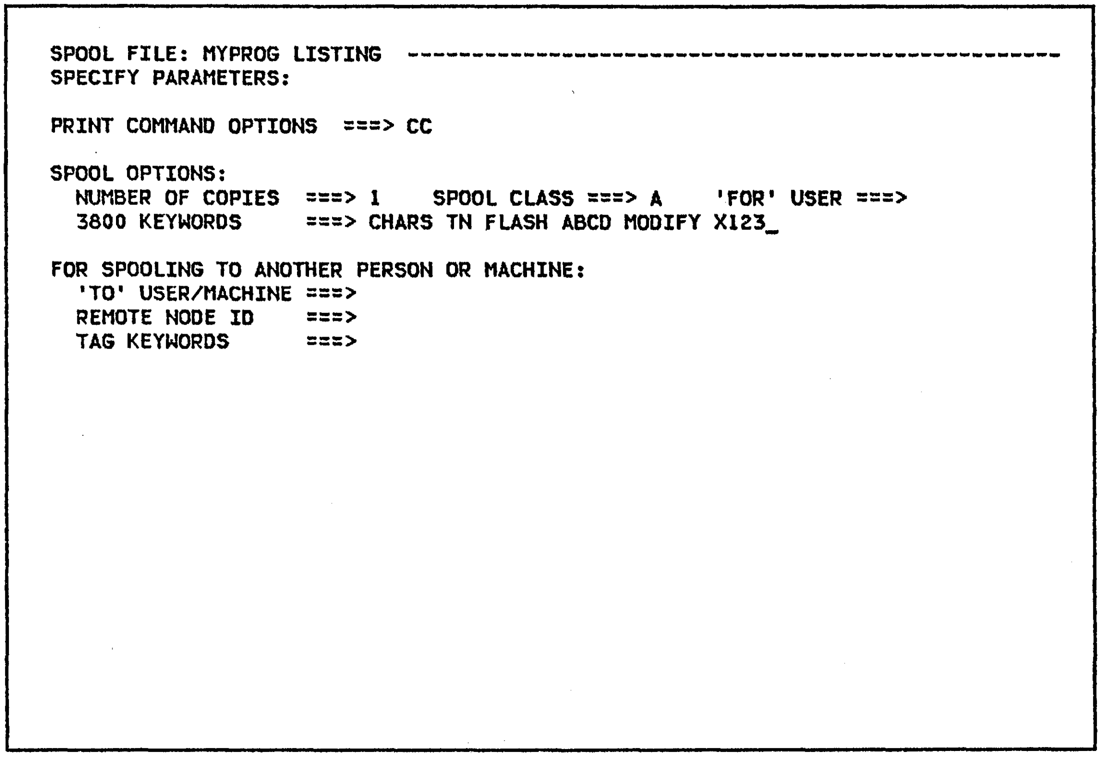
- Remote node id - if the desination is a remote node on the network (valid with either "to user/machine" or "tag keywords", but not both).

- Tag keywords - to specify control parameters for printing on a remote non-VM system. See description of TAG command in either the VM/370 Networking manual, SH20-1977, or the VM/370 RSCS Networking manual, SH24-5005.

If "other parms - no" was specified on the first menu, the second menu is not displayed and the remembered information is not used.

SPF will generate a PRINT, PUNCH, or DISK DUMP command (depending on the option selected) to spool the data to the specified desination. The first menu is then redisplayed to allow specification of additional data to be spooled. If a different option and/or different spool parameters are specified, the spool file is closed (releasing the data for output) prior to generating the next PRINT, PUNCH, or DISK DUMP command. When the End PF key is pressed any remaining spooled data is released for output, and the user is returned to the next higher-level menu.

Caution: Use of this utility in split screen mode should be avoided if the other logical screen is also being used to generate output for the virtual printer or punch. The data will be intermixed on the printer or punch if the user jumps back and forth between the two screens. No data will be lost, but the results may be confusing.



```
SPOOL FILE: MYPROG LISTING ---------------------------------------------------
SPECIFY PARAMETERS:

PRINT COMMAND OPTIONS  ===> CC

SPOOL OPTIONS:
  NUMBER OF COPIES   ===> 1    SPOOL CLASS ===> A    'FOR' USER ===>
  3800 KEYWORDS      ===> CHARS TN FLASH ABCD MODIFY X123_

FOR SPOOLING TO ANOTHER PERSON OR MACHINE:
  'TO' USER/MACHINE ===>
  REMOTE NODE ID    ===>
  TAG KEYWORDS      ===>
```

Figure 60. Spool utility - second menu

READER UTILITY (OPTION 3.7)

The reader utility allows the user to retrieve files that are on his virtual reader, and is especially intended to be used to retrieve output (TEXT files, LISTING files, etc.) produced by the CMS batch machine as a result of SPF batch processing (option 5).

The options which are displayed when this utility is selected (utility option 7) are dependent upon the status and characteristics of the first active file on the user's virtual reader.

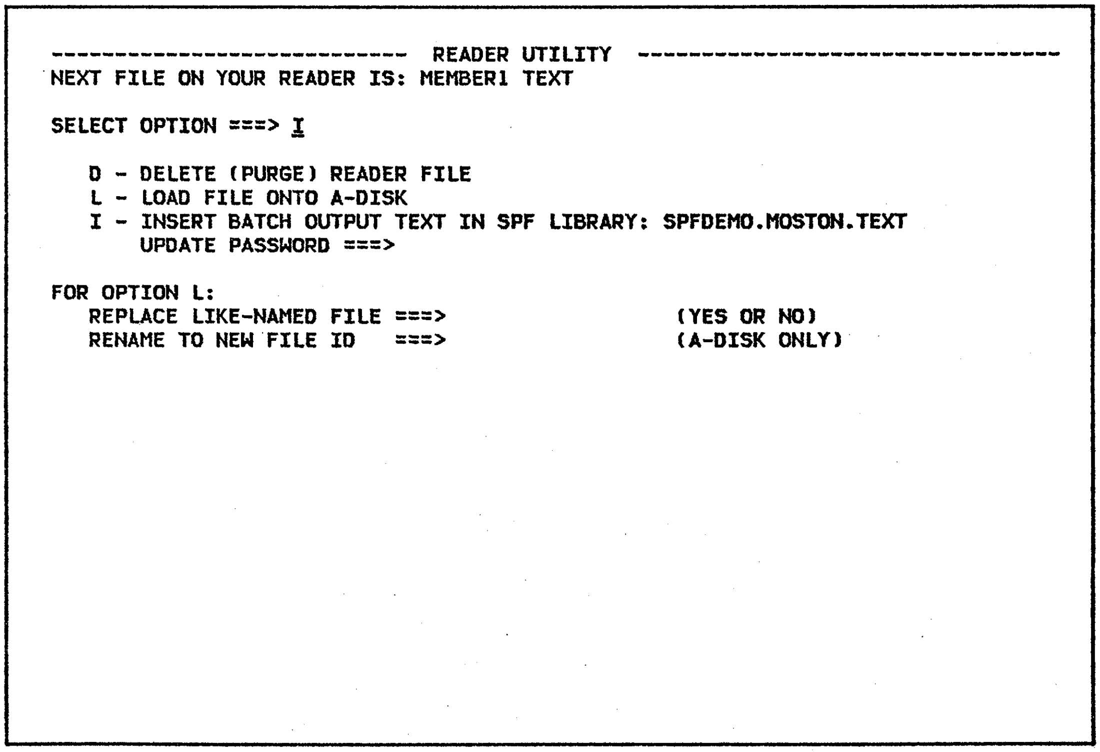
If the first file on the reader is a TEXT file that was produced by an SPF batch compile or assembly (option 5), and the source was from an SPF library, the options are:

D - Delete (purge) reader file
L - Load file onto A-disk
I - Insert batch output TEXT in SPF library

An example of this menu is shown in Figure 61. The default option in this case is "I" (pre-entered on the menu). If option "I" is used, the user must specify an update password if the SPF TEXT library (into which the file is to be inserted) is on a password protected minidisk.

If the user selects option "L", the following additional information may be specified:

- Replace like-named file (yes/no) - where "no" prevents a file with the same filename and filetype on the user's A-disk from being overwritten.
- Rename to new file id - which allows the file in the virtual reader to be renamed (new filename and/or filetype) in the process of moving it to the user's A-disk.



```
----------------------------  READER UTILITY  --------------------------------
NEXT FILE ON YOUR READER IS: MEMBER1 TEXT

SELECT OPTION ===> I

   D - DELETE (PURGE) READER FILE
   L - LOAD FILE ONTO A-DISK
   I - INSERT BATCH OUTPUT TEXT IN SPF LIBRARY: SPFDEMO.MOSTON.TEXT
       UPDATE PASSWORD ===>

FOR OPTION L:
   REPLACE LIKE-NAMED FILE ===>                 (YES OR NO)
   RENAME TO NEW FILE ID   ===>                 (A-DISK ONLY)
```

Figure 61. Reader utility menu

The other menus displayed by this utility are similar to the one shown in Figure 61, except as follows:

1. If the first file on the reader is not a TEXT file for which the source was from an SPF library, the "I" option (and associated password field) will not be displayed. if the file was "disk dumped" to the virtual reader, the second option on the menu will be:

   L - Load file onto A-disk

   If the file was "punched" to the virtual reader, the second option on the menu will be:

   R - Readcard file to A-disk

   SPF automatically determines whether the file was disk dumped or punched.

   The second option ("L" or "R") will be pre-entered in the option field. The replace (yes/no) and rename parameters also apply to these menus for options "L" or "R".

2. If the reader is empty, or if the first file in the reader cannot be processed by this utility, one of the following messages will be displayed on the second line of the menu:

   ```
   NEXT FILE ON YOUR READER IS: READER IS EMPTY
   NEXT FILE ON YOUR READER IS: NOT CLASS A FILE
   NEXT FILE ON YOUR READER IS: FILE IN HOLD STATUS
   ```

   In these cases, the only option is to press the End PF key to return to the previous menu. For files which are not class A or which are in hold status, a CMS command may be entered (under SPF option 6) to change the class or status. This utility may then be reselected to continue processing the virtual reader.

When the user selects an option and presses the ENTER key, the first file on the user's virtual reader is processed. SPF then examines the next file on the reader and redisplays the appropriate variant of the reader utility menu, so that the next file can be processed. This continues until the user presses the End PF key to return to the next higher level menu.

RETRIEVE UTILITY (OPTION 3.8)

When this option is selected (utility option 8), a menu is displayed which allows the user to specify a tape unit and serial number to retrieve SPF libraries (via tape) from a TSO system (Figure 62). The tape drive should be attached to the user's machine and readied before this utility is used.

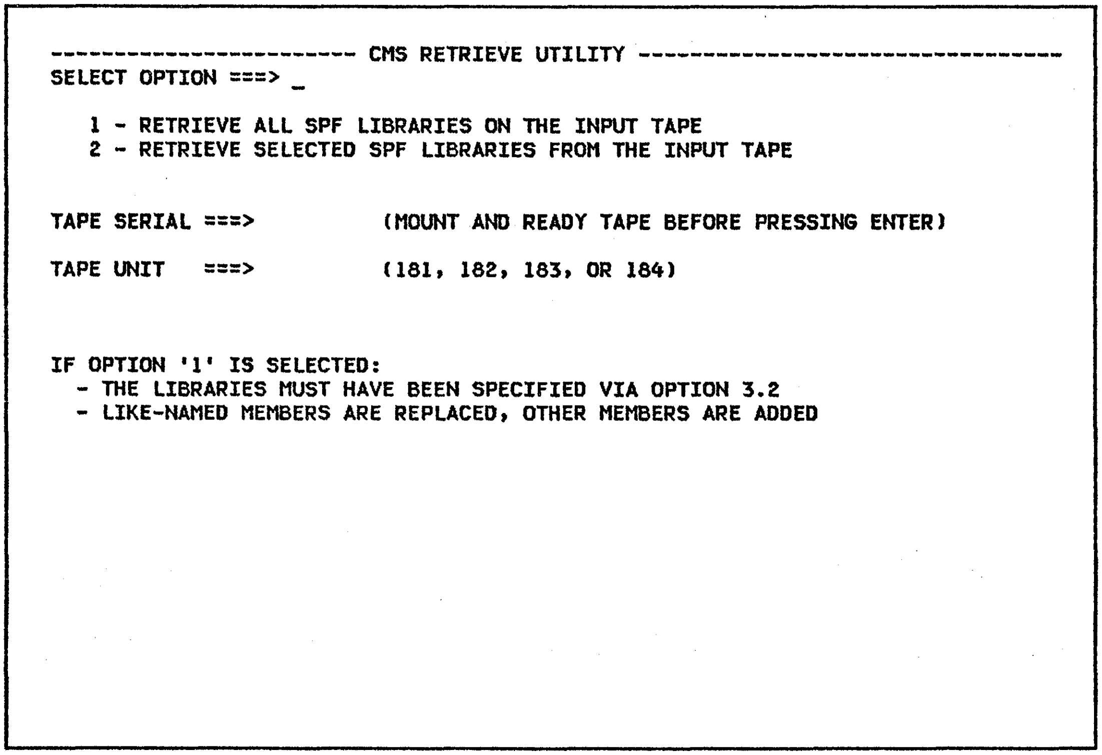
This utility has the following options:

1 - Retrieve all SPF libraries on the input tape
2 - Retrieve selected SPF libraries from the input tape

The tape must be a standard labeled tape created by the IEHMOVE utility. Any SPF/TSO library (partitioned data set with a 3-level name) with fixed or variable record format can be processed. SPF library statistics are also retrieved during the processing.

Note: This utility may not be used to place members into an SPF/CMS TEXT library with organization "T" (TXTLIB).

When the end of tape is reached, or the End or Return PF key is pressed, the tape is rewound but not unloaded.



```
------------------------ CMS RETRIEVE UTILITY ---------------------------------
SELECT OPTION ===> _

   1 - RETRIEVE ALL SPF LIBRARIES ON THE INPUT TAPE
   2 - RETRIEVE SELECTED SPF LIBRARIES FROM THE INPUT TAPE


TAPE SERIAL ===>          (MOUNT AND READY TAPE BEFORE PRESSING ENTER)

TAPE UNIT   ===>          (181, 182, 183, OR 184)


IF OPTION '1' IS SELECTED:
  - THE LIBRARIES MUST HAVE BEEN SPECIFIED VIA OPTION 3.2
  - LIKE-NAMED MEMBERS ARE REPLACED, OTHER MEMBERS ARE ADDED
```

Figure 62. Retrieve utility menu

Retrieve All Libraries

For option "1" the libraries must have been previously specified to SPF/CMS via utility 3.2 ("specify" option). Project name, library name, and type for each SPF library on tape must exactly match an SPF/CMS library. Tape libraries for which no match is found are skipped.

The record length of the receiving library must be equal to or greater than that of the tape library, and the record formats must agree. Blocked record formats are accepted.

The receiving library may be empty, or it may already contain members. As members are copied from the tape into the specified library, new members are added and like-named members are replaced.

Retrieve Selected Libraries

For option "2" the identification of each SPF library encountered on tape may be reviewed at the terminal as the tape is read. The user may then choose to retrieve the library, skip to the next library on the tape, or search the tape for a specified library. Optionally, a library may be renamed as it is retrieved.

If a library has not been previously specified to SPF/CMS, its characteristics may be specified during the retrieval process. For libraries which have already been specified, the rules for record format and length are the same as for option "1", but the replacement of like-named members is under user control.

Either of two secondary menus will be displayed for each library read from tape (see Figure 63). The first menu shown in the figure will be displayed if the library is already specified to SPF/CMS. The user then has the following options:

M - Merge members from tape with the specified SPF/CMS library. For this option; the user must specify the yes/no option for replacement of like-named members. (The library may be specified and empty.)

I - Identify a new library name (rename option). For this option, a new library (project name, library name, and type) must be specified. The appropriate secondary menu will then be displayed (or redisplayed), based on whether the new library is already specified to SPF/CMS, and the user may again choose the desired option.

S - Skip to the next library on the tape. A secondary menu will be displayed for the next library, and the user may choose the desired option.

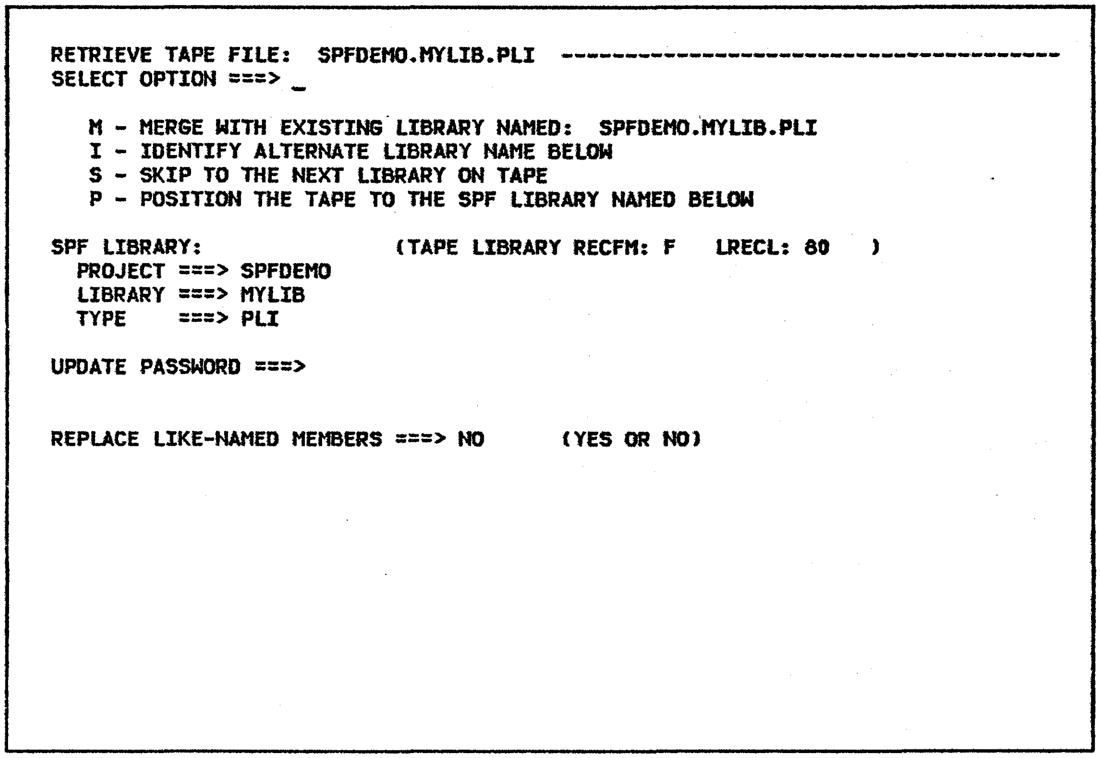
P - Position tape to a specified library (and skip the current library). For this option, the user must supply the project name, library name, and type for a library on the tape. If the library is found a secondary menu will be displayed, and the user may choose the desired option.

If the end of tape is encountered before the library is found, the tape is rewound and the first retrieve menu is redisplayed. If the tape is positioned beyond the specified library, it will be backspaced to that library.

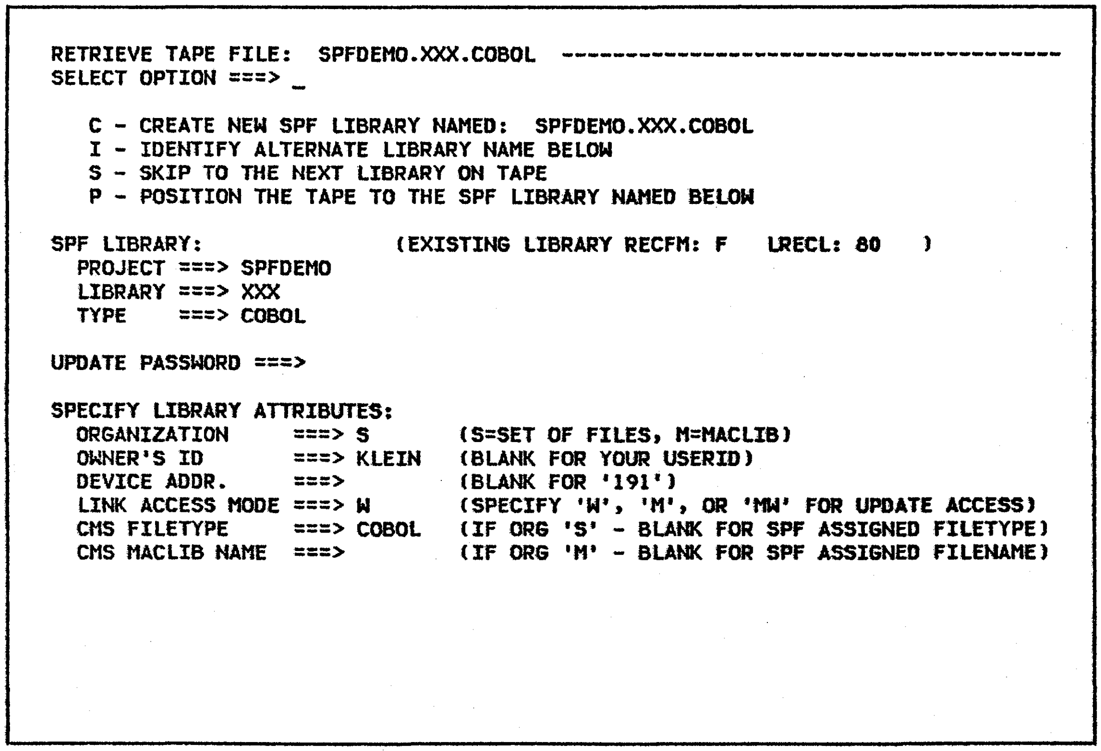
If the library read from tape has not been specified to SPF/CMS, the second menu shown in Figure 63 will be displayed. The first option on this menu is:

C - Create new SPF library. For this option, the library attributes must be specified on the lower part of the menu. This is the same information which may be specified under the file utility, option 3.2 ("specify"), except that the record format and length are are taken from the tape library.

The other options on this menu are the same as those described above.



```
RETRIEVE TAPE FILE:  SPFDEMO.MYLIB.PLI  --------------------------------------
SELECT OPTION ===> _

   M - MERGE WITH EXISTING LIBRARY NAMED:  SPFDEMO.MYLIB.PLI
   I - IDENTIFY ALTERNATE LIBRARY NAME BELOW
   S - SKIP TO THE NEXT LIBRARY ON TAPE
   P - POSITION THE TAPE TO THE SPF LIBRARY NAMED BELOW

SPF LIBRARY:                  (TAPE LIBRARY RECFM: F   LRECL: 80    )
  PROJECT ===> SPFDEMO
  LIBRARY ===> MYLIB
  TYPE    ===> PLI

UPDATE PASSWORD ===>


REPLACE LIKE-NAMED MEMBERS ===> NO      (YES OR NO)
```



```
RETRIEVE TAPE FILE:  SPFDEMO.XXX.COBOL  --------------------------------------
SELECT OPTION ===> _

   C - CREATE NEW SPF LIBRARY NAMED:  SPFDEMO.XXX.COBOL
   I - IDENTIFY ALTERNATE LIBRARY NAME BELOW
   S - SKIP TO THE NEXT LIBRARY ON TAPE
   P - POSITION THE TAPE TO THE SPF LIBRARY NAMED BELOW

SPF LIBRARY:                  (EXISTING LIBRARY RECFM: F   LRECL: 80    )
  PROJECT ===> SPFDEMO
  LIBRARY ===> XXX
  TYPE    ===> COBOL

UPDATE PASSWORD ===>

SPECIFY LIBRARY ATTRIBUTES:
  ORGANIZATION     ===> S       (S=SET OF FILES, M=MACLIB)
  OWNER'S ID       ===> KLEIN   (BLANK FOR YOUR USERID)
  DEVICE ADDR.     ===>         (BLANK FOR '191')
  LINK ACCESS MODE ===> W       (SPECIFY 'W', 'M', OR 'MW' FOR UPDATE ACCESS)
  CMS FILETYPE     ===> COBOL   (IF ORG 'S' - BLANK FOR SPF ASSIGNED FILETYPE)
  CMS MACLIB NAME  ===>         (IF ORG 'M' - BLANK FOR SPF ASSIGNED FILENAME)
```

Figure 63. Retrieve utility - secondary menus

SCRIPT/VS UTILITY (OPTION 3.9)

When this option is selected (utility option 9), a sequence of menus is displayed to allow formatting, displaying, and printing of text maintained in SPF libraries or sequential CMS files. Use of this utility requires the installation of the Document Composition Facility (SCRIPT/VS), program product 5748-XX9, with the Foreground Environment Feature.

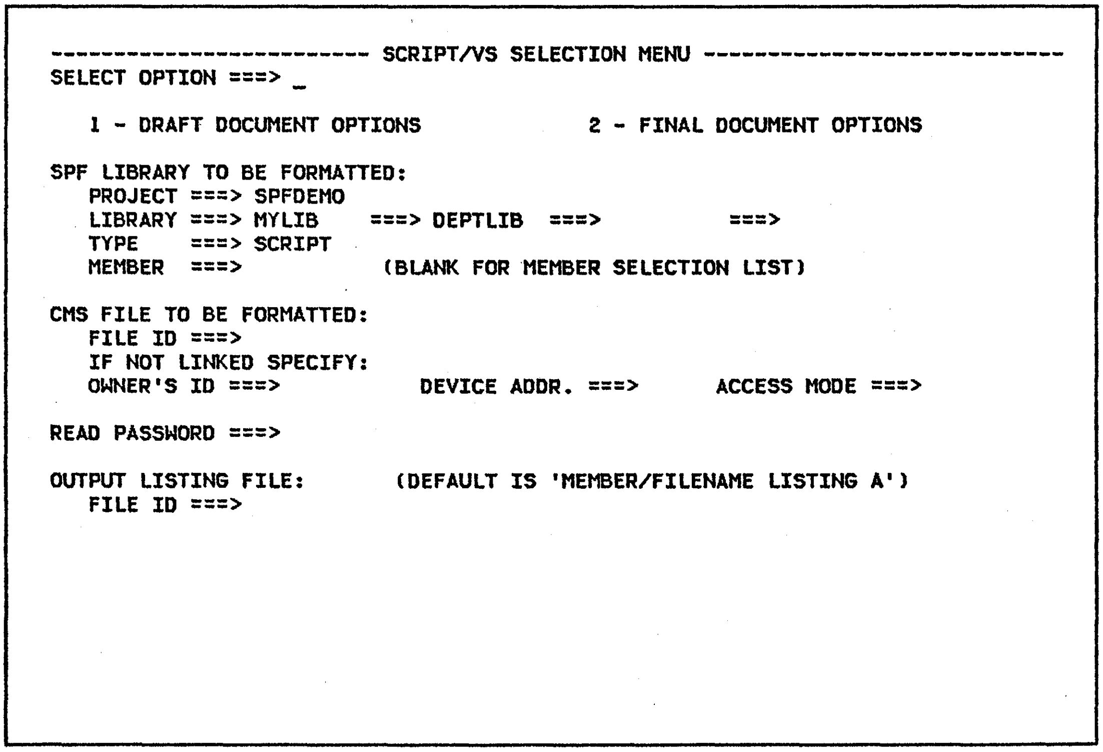
This utility has the following options:

1 - Draft document options
2 - Final document options

In the draft document option, it is assumed that the primary interest of the user is to assure that the content, syntax, and spelling within the document are correct. The final document option provides additional flexibility and control for the user interested in specific document formats.

The SCRIPT/VS selection menu is used to select the draft or final document option, and to specify the library or file to be formatted (Figure 64). For SPF libraries, if a member name is not specified, a member list will be displayed. A concatenated sequence of SPF libraries may be specified to allow inclusion of imbedded or appended members from more than one library.

Any SPF libraries to be processed by this utility must have an organization of "S" (set of sequential files). They need not have an SPF type qualifier of SCRIPT, and the corresponding set of sequential files need not have a filetype of SCRIPT. If, however, a CMS file id is specified (on the second part of the menu) it must have a filetype of SCRIPT, and any imbedded (or appended) files must also have a filetype of SCRIPT unless they are explicitly defined via a ".dd" control word.



```
------------------------- SCRIPT/VS SELECTION MENU ----------------------------
SELECT OPTION ===> _

   1 - DRAFT DOCUMENT OPTIONS            2 - FINAL DOCUMENT OPTIONS

SPF LIBRARY TO BE FORMATTED:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB     ===> DEPTLIB  ===>          ===>
   TYPE    ===> SCRIPT
   MEMBER  ===>           (BLANK FOR MEMBER SELECTION LIST)

CMS FILE TO BE FORMATTED:
   FILE ID ===>
   IF NOT LINKED SPECIFY:
   OWNER'S ID ===>          DEVICE ADDR. ===>      ACCESS MODE ===>

READ PASSWORD ===>

OUTPUT LISTING FILE:       (DEFAULT IS 'MEMBER/FILENAME LISTING A')
   FILE ID ===>
```

Figure 64. SCRIPT/VS selection menu

Processing Sequence

The processing sequence for the SCRIPT/VS utility is as follows:

1. The user enters the appropriate information on the SCRIPT/VS selection menu, and presses the ENTER key.

2. A secondary menu is displayed for either the "draft" or "final" document options, depending on which option was specified on the first menu. The user enters the appropriate information on the secondary menu (see below) and presses the ENTER key.

3. The SCRIPT/VS formatter is invoked. If the "quiet" option has not been specified, the screen is blanked and a one-line message is displayed, indicating that SCRIPT/VS processing has started. When processing im complete, a "MORE..." message is displayed in the lower right-hand corner of the screen. The user then presses PA2.

4. The formatted output is displayed in browse mode. The output may be scrolled using the Scroll PF keys. All browse commands may be entered. When the user has finished browsing the output, he presses the End PF key.

5. The SCRIPT/VS print menu is then displayed to allow the formatted output to be printed, kept, or deleted. The user enters the appropriate information on this menu (see below) and presses the ENTER key.

6. The specified action (print, keep, or delete) is taken and the SCRIPT/VS selection menu is redisplayed. The user may then format additional documents, or press the End PF key to return to a higher level menu.

Draft Document Options

When option 1 is selected on the SCRIPT/VS selection menu, a secondary menu is displayed to allow the user to specify the draft document options (Figure 65). All the parameters on this menu are optional.

For automatic spelling verification, "yes" is entered in the spelling check field. Spelling verification must also be enabled within the document. (See the ".sv" command in the SCRIPT/VS User's Guide.)

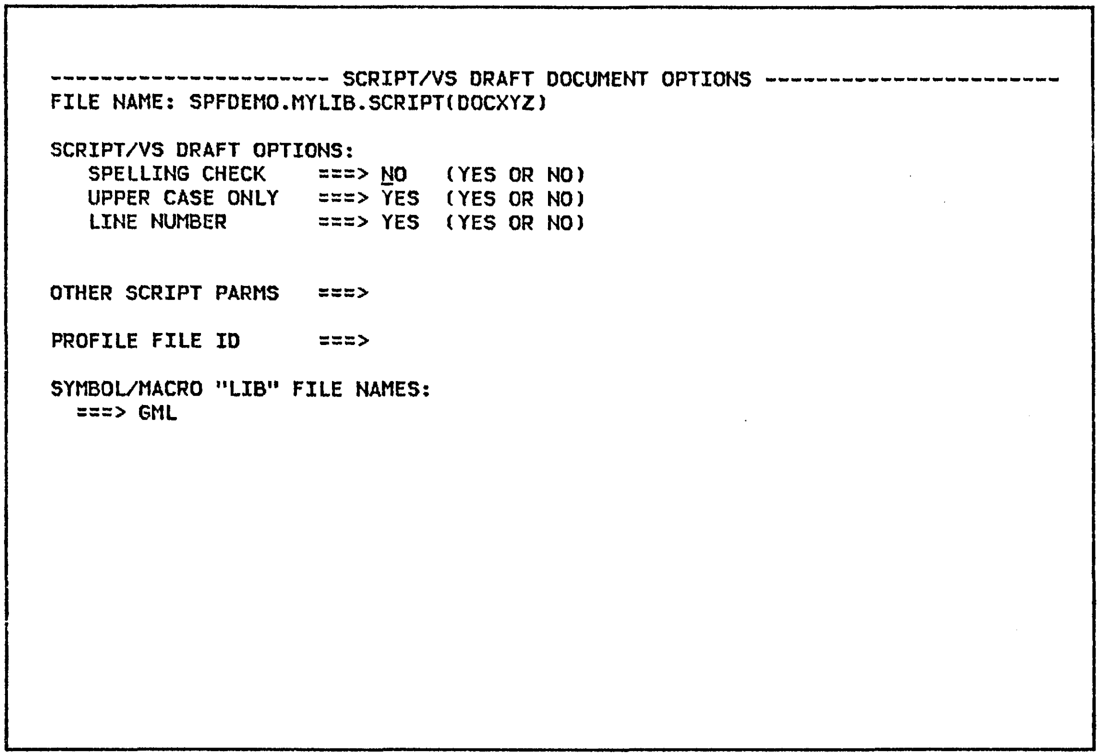
For all upper case printing, "yes" is entered in the upper case only field.

For line numbers to be printed in the output document, "yes" is entered in the line number field. Line numbers can be useful for correcting errors and mis-spelled words, which SCRIPT/VS identifies by line number.

Use of the "other SCRIPT parms" field allows input of any SCRIPT/VS parameters not on the menu. The NOPROF option should not be specified if the source is from an SPF library.

If a SCRIPT/VS document profile is to be used, the user must enter the name of the CMS file which contains the profile in the "profile" field. A file id of PROFILE SCRIPT should be entered for the default profile.

To incorporate one or more user GML library in formatting the data, the user must enter the filename(s) in the "symbol/macro lib" field. These files must be MACLIBs. If no files are specified, the default GML library, if any, will be used.



```
---------------------- SCRIPT/VS DRAFT DOCUMENT OPTIONS ----------------------
FILE NAME: SPFDEMO.MYLIB.SCRIPT(DOCXYZ)

SCRIPT/VS DRAFT OPTIONS:
   SPELLING CHECK    ===> NO   (YES OR NO)
   UPPER CASE ONLY   ===> YES  (YES OR NO)
   LINE NUMBER       ===> YES  (YES OR NO)


OTHER SCRIPT PARMS   ===>

PROFILE FILE ID      ===>

SYMBOL/MACRO "LIB" FILE NAMES:
  ===> GML
```

Figure 65. SCRIPT/VS draft document options menu

Final Document Options

When option 2 is selected on the SCRIPT/VS selection menu, a secondary menu is displayed to allow the user to specify the final document options (Figure 66). All the parameters on this menu are optional. However, if a device type is entered, the lines per inch and form definition fields must be correctly filled in.

For automatic spelling verification, "yes" is entered in the spelling check field. Spelling verification must also be enabled within the document. (See the ".sv" command in the SCRIPT/VS User's Guide.)

For all upper case printing, "yes" is entered in the upper case only field.

The next four fields on the menu are used to describe the output device and the physical characteristics of the printed page. The type of printer on which the output will be printed (and for which SCRIPT/VS will format the document) is specified in the "device type" field. If the document is to be printed on a device other than an IBM 1403 or 3800, this field should be left blank, and the printer should be specified in the "other SCRIPT parms" field (see below).

The "lines/inch", "form width" and "form length" fields are used to describe how the document should physically fit on the printed page.

The "3800 chars parm" field is used to specify the type font(s) if the the document is being formatted for an IBM 3800 printer. The user may enter one or two SCRIPT/VS fonts or up to four 3800 character arrangement tables.

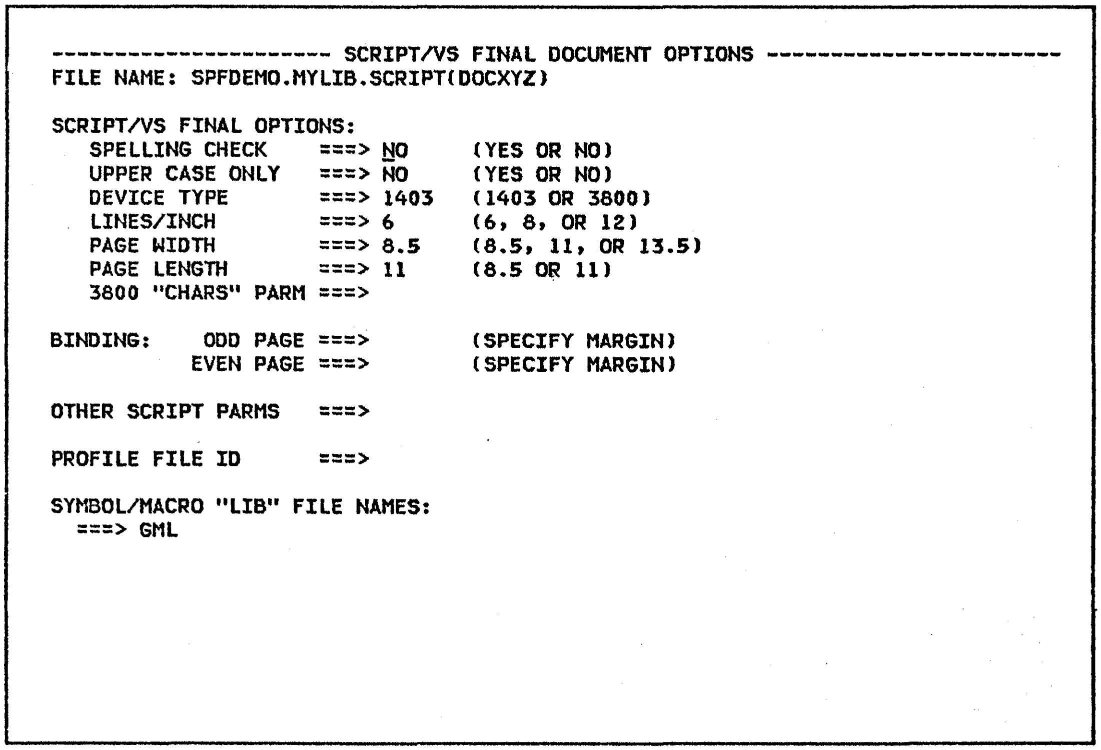
The user may specify the page margin to be used in binding the document by entering the size of the margin in the "binding" fields. A different margin can be specified for even and odd numbered pages. For more information, see "BIND" in the SCRIPT/VS User's Reference Guide.

Use of the "other SCRIPT parms" field allows input of any SCRIPT/VS parameters not on the menu. The NOPROF option should not be specified if the source is from an SPF library.

If the "other SCRIPT parms" field is used to specify a device other than the 1403 or 3800, the device must have been previously added to the SCRIPT/VS device characteristics table. A device may be selected from this table by leaving the "device type" field of the menu blank and specifying the SCRIPT/VS device parameter in the "other SCRIPT parms" field.

If a SCRIPT/VS document profile is to be used, the user must enter the name of the CMS file which contains the profile in the "profile" field. A file id of PROFILE SCRIPT should be entered for the default profile.

To incorporate one or more user GML library in formatting the data, the user must enter the filename(s) in the "symbol/macro lib" field. These files must be MACLIBs. If no files are specified, the default GML library, if any, will be used.



```
---------------------- SCRIPT/VS FINAL DOCUMENT OPTIONS ----------------------
FILE NAME: SPFDEMO.MYLIB.SCRIPT(DOCXYZ)

SCRIPT/VS FINAL OPTIONS:
   SPELLING CHECK     ===> NO      (YES OR NO)
   UPPER CASE ONLY    ===> NO      (YES OR NO)
   DEVICE TYPE        ===> 1403    (1403 OR 3800)
   LINES/INCH         ===> 6       (6, 8, OR 12)
   PAGE WIDTH         ===> 8.5     (8.5, 11, OR 13.5)
   PAGE LENGTH        ===> 11      (8.5 OR 11)
   3800 "CHARS" PARM  ===>

BINDING:      ODD PAGE ===>        (SPECIFY MARGIN)
             EVEN PAGE ===>        (SPECIFY MARGIN)

OTHER SCRIPT PARMS    ===>

PROFILE FILE ID       ===>

SYMBOL/MACRO "LIB" FILE NAMES:
  ===> GML
```

Figure 66. SCRIPT/VS final document options menu

SCRIPT/VS Print Options

When the user has finished browsing the formatted document, pressing the End PF key will cause the SCRIPT/VS print menu to be displayed (Figure 67). On this menu, the "file id" field shows the id of the output listing file, as specified on the SCRIPT/VS selection menu. This file contains the formatted text to be processed (print and delete, delete, or keep) as specified on this menu.

If option "P" (print file) is selected, the following spool parameters may be specified:

- Number of copies - if more than one copy is desired.

- Spool class - for other than class A.

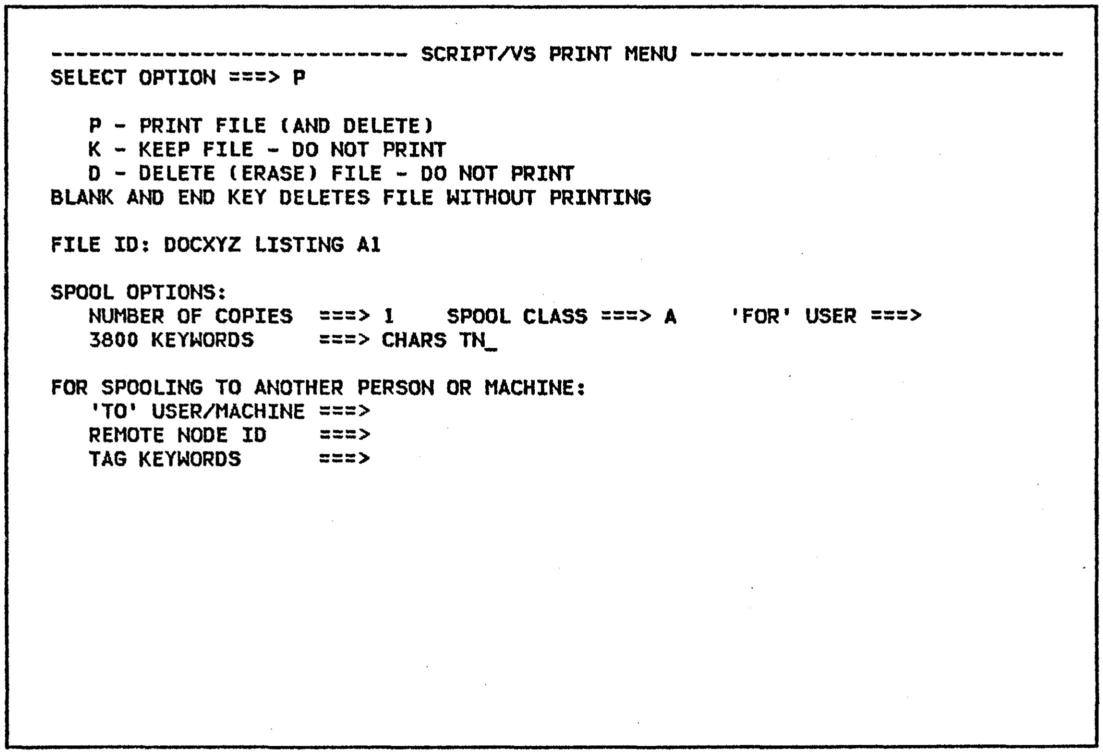
- For user - to spool the output to another user's virtual printer on the same VM system (ignored if either "to user/machine" or "remote node id" is specified).

- 3800 keywords - if output is directed to an IBM 3800 printer which is attached to the CMS system as a virtual spooling device. See description of SPOOL command in the VM/370 CP Command Reference, GC20-1820.

- To user/machine - to spool the output to the virtual reader of another user or machine.

- Remote node id - if the desination is a remote node on the network (valid with either "to user/machine" or "tag keywords", but not both).

- Tag keywords - to specify control parameters for printing on a remote non-VM system. See description of TAG command in either the VM/370 Networking manual, SH20-1977, or the VM/370 RSCS Networking manual, SH24-5005.



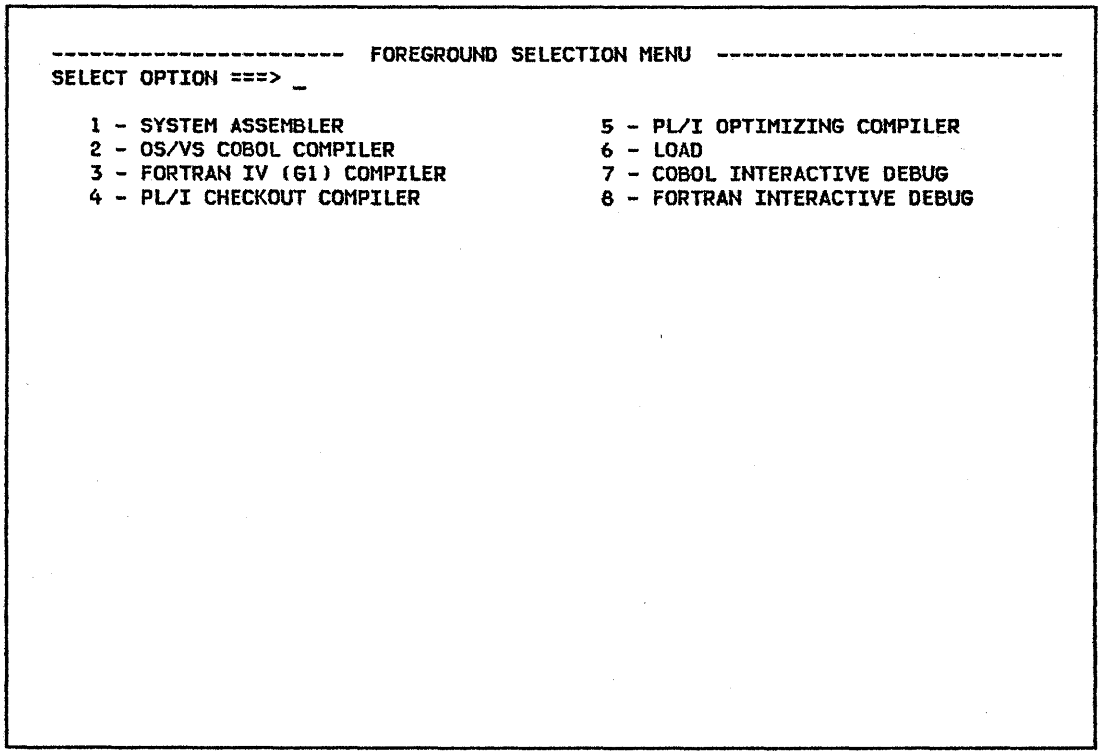
```
----------------------------- SCRIPT/VS PRINT MENU -----------------------------
SELECT OPTION ===> P

   P - PRINT FILE (AND DELETE)
   K - KEEP FILE - DO NOT PRINT
   D - DELETE (ERASE) FILE - DO NOT PRINT
BLANK AND END KEY DELETES FILE WITHOUT PRINTING

FILE ID: DOCXYZ LISTING A1

SPOOL OPTIONS:
   NUMBER OF COPIES  ===> 1    SPOOL CLASS ===> A    'FOR' USER ===>
   3800 KEYWORDS     ===> CHARS TN_

FOR SPOOLING TO ANOTHER PERSON OR MACHINE:
   'TO' USER/MACHINE ===>
   REMOTE NODE ID    ===>
   TAG KEYWORDS      ===>
```

Figure 67. SCRIPT/VS print menu

FOREGROUND PROCESSING (OPTION 4)

The foreground option provides an interface with standard language processors for foreground compilation and assembly of programs stored in SPF libraries or CMS files. It also provides a menu interface to the FORTRAN and COBOL interactive debug facilities, and to the CMS LOAD command.

When foreground is selected (primary option 4), a secondary menu is displayed to allow the user to select a language processor (Figure 68). It is possible to bypass the secondary menu by entering two numbers, separated with a decimal point, on the primary option menu. For example, entering "4.3" on the primary menu has the same effect as entering "4" on the primary menu and "3" on secondary menu.

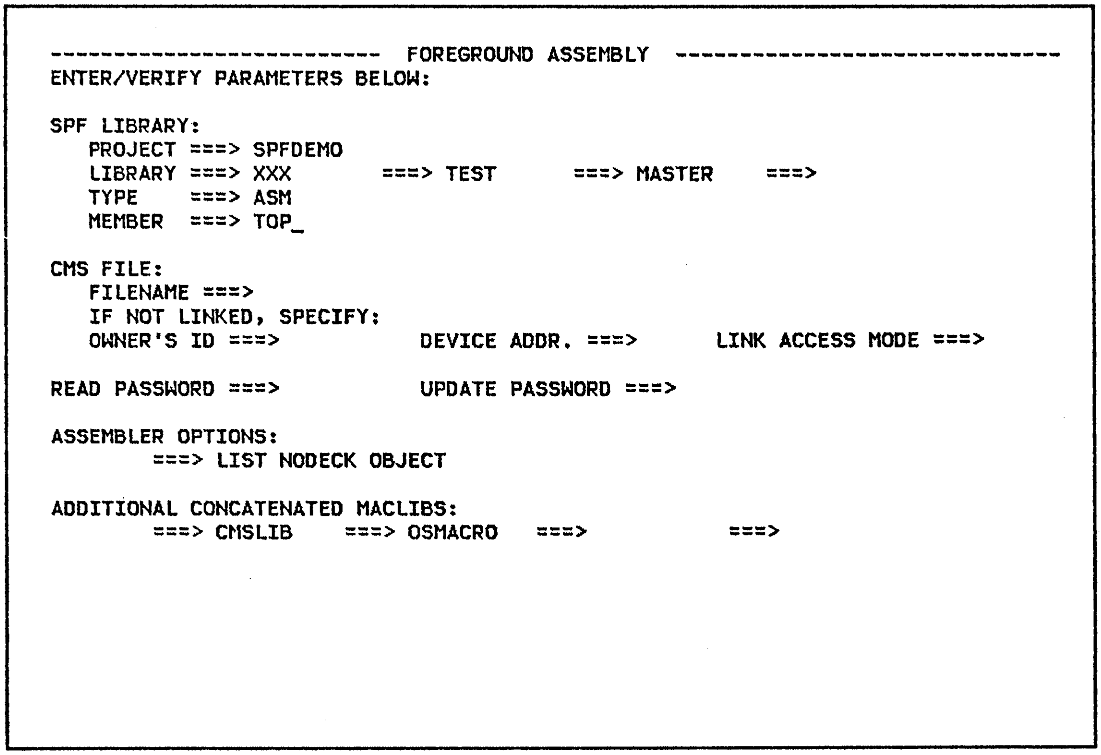


```
------------------------ FOREGROUND SELECTION MENU ---------------------------
SELECT OPTION ===> _

   1 - SYSTEM ASSEMBLER                      5 - PL/I OPTIMIZING COMPILER
   2 - OS/VS COBOL COMPILER                  6 - LOAD
   3 - FORTRAN IV (G1) COMPILER              7 - COBOL INTERACTIVE DEBUG
   4 - PL/I CHECKOUT COMPILER                8 - FORTRAN INTERACTIVE DEBUG
```

Figure 68. Foreground selection menu

When a language processor has been selected, a menu is displayed to allow the user to specify SPF library information or CMS filename, and processor options. Figure 69 shows an example for Assembler. The project name, first library name, type qualifier, and member name must be specified to assemble a program whose source is in an SPF library. The filename must be specified to assemble a CMS file with a filetype of ASSEMBLE. (If the CMS filename is entered, it overrides any SPF library specification.) The other parameters are optional, except as noted below.

Once the processor menu has been filled in and the ENTER key has been pressed, the appropriate language processor is invoked. The GLOBAL command and the command that invokes the language processor will be displayed, as will certain other output of the language processor (depending on the options chosen), such as error messages, prompts, and return codes. Any communication with the language processor is in line I/O mode.



```
--------------------------- FOREGROUND ASSEMBLY -----------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX        ===> TEST       ===> MASTER     ===>
   TYPE    ===> ASM
   MEMBER  ===> TOP_

CMS FILE:
   FILENAME ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>           UPDATE PASSWORD ===>

ASSEMBLER OPTIONS:
        ===> LIST NODECK OBJECT

ADDITIONAL CONCATENATED MACLIBS:
        ===> CMSLIB    ===> OSMACRO   ===>           ===>
```

Figure 69. Foreground - Assembler example

Upon completion of the foreground processing, a "MORE..." message will be displayed in the lower right-hand corner of the screen. The user may then press the PA2 key to return to the foreground processor menu.

When the foreground processor menu is redisplayed, a message indicating completion of the process is displayed in the upper right-hand corner of the screen. The user may then enter another foreground processing request or press the End PF key to return to the previous menu.

ASSEMBLER/COMPILER PROCESSING

The Assembler menu shown in Figure 69 is typical of the language processor menus. In the figure, a library concatenation sequence of three data sets has been specified. The concatenation order is:

    SPFDEMO.XXX.ASM
    SPFDEMO.TEST.ASM
    SPFDEMO.MASTER.ASM

Note: The SPF type qualifier for an assembler source library need not be ASSEMBLE, as illustrated in the example.

Before invoking the foreground processor, SPF will scan the concatenated sequence of libraries to find the member to be assembled or compiled (member TOP in this example). A CMS file with a filetype acceptable to the language processor (ASSEMBLE in this example) will be created on the user's A-disk by extracting the member from the first library in the sequence in which it occurs.

For any library in the sequence that is a collection of sequential files, a MACLIB is created on the user's A-disk to enable the use of COPY or INCLUDE statements. (For performance reasons, it is recommended that high-level libraries be specified as MACLIBs.)

Any temporary files created by SPF will be erased at the completion of the foreground process.

For this example, if SPFDEMO.XXX.ASM is a colletion of sequential files and the higher-level libraries are already MACLIBs, the GLOBAL command preceding the ASSEMBLE command might look like:

```
GLOBAL MACLIB SPF00511 TEST MASTER CMSLIB OSMACRO
```

where SPF00511 is the SPF generated filename for the MACLIB containing all the members of SPFDEMO.XXX.ASM.

Note the following points:

1. The concatenation sequence serves two purposes: It is used by SPF to locate the primary member to be assembled, and it allows inclusion of subsidiary members referenced by COPY statements in the assembler source.

2. Other MACLIBs can be appended to the GLOBAL statement, following the SPF libraries, by specifying them in the desired order in the "additional concatenated MACLIBs" fields.

3. If the source is from an SPF library, the TEXT file is placed in an SPF library with a three-level name composed of the project name, the first library name, and a type qualifier of TEXT. The SPF member name for the TEXT file is the same as the source member. The LISTING file is placed on the user's A-disk. Its filename is the same as the source member name.

4. If the source is specified as a CMS file (rather than an SPF library), the filetype must be whatever is acceptable to the language processor (ASSEMBLE in this example). The resulting TEXT and LISTING files are placed on the user's A-disk.

5. In specifying a CMS file that would require a LINK and ACCESS, the user must provide the necessary LINK information: The user id of the owner, the virtual device address of the minidisk on which the source resides, and the desired link access mode (e.g, RR).

6. The language processor options are passed to the processor exactly as specified by the user. Up to 16 tokens may be specified as processor options. Only those options specified on the menu are passed to the language processor; SPF does not automatically generate any options.

The processor menus for COBOL, FORTRAN, and the two PL/I compilers are similar to the one for Assembler. The meanings of the options differ among the several language processors. Check the user's guide for the specific language for an explanation of the processor options.

Compiler output is handled differently in the following cases:

- When the TEST option is specified for the FORTRAN compiler, a special TEXT file is produced for later use by the FORTRAN interactive debug facility. The file is not placed in an SPF library. Rather, it is left on the user's A-disk, as is the generated FORTRAN file.

- When the TEST option is specified for the COBOL compiler, a SYSUT5 file is produced for later use by the COBOL interactive debug facility. The file is placed on the user's A-disk.

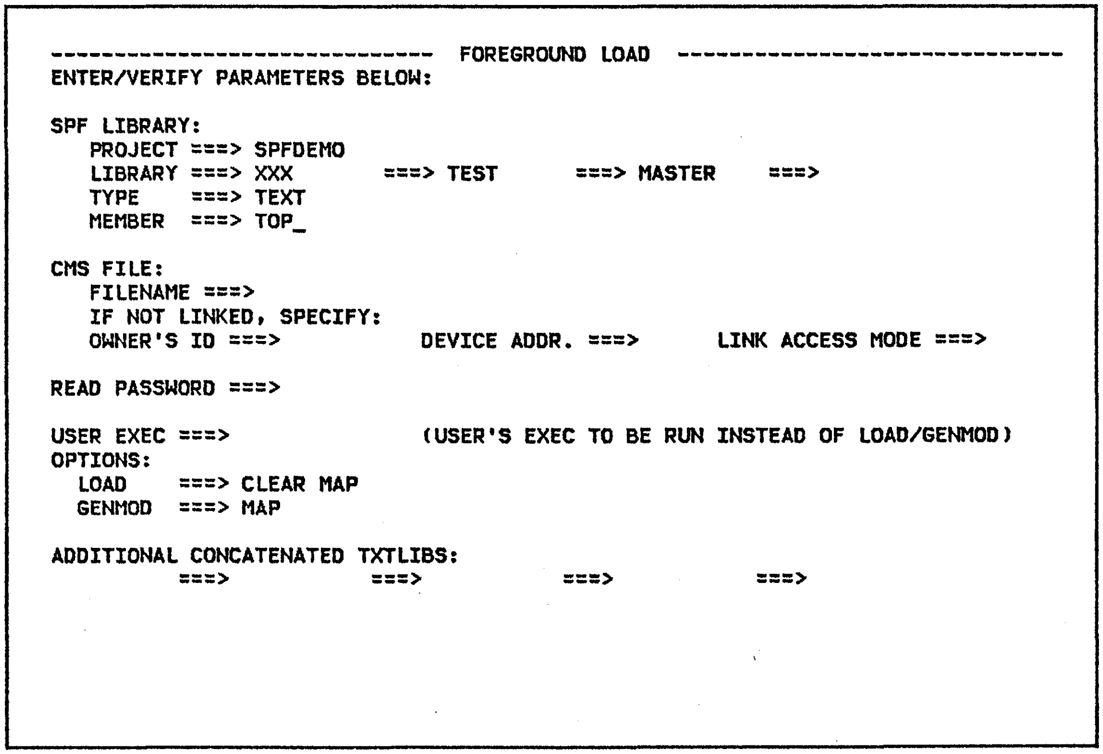
- When the OBJECT option is specified for the PL/I Checkout compiler, a special TEXT file (a "link edit stub") and an ITXTLIB file (interpretable text) are produced for later use with the PLICR command. Both files are placed on the user's A-disk.

The menus for the interactive debug facilities differ somewhat from the other foreground menus:

- The COBOL interactive debug menu allows specification of up to four additional program ids (which must match the external filenames), and PRINT and SOURCE options which determine what FILEDEFs are generated by SPF. Also allowed are 15 tokens of execution parameters that are stacked for use by the processor, and the specification of up to four additional TXTLIBs.
- The FORTRAN interactive debug menu allows variations of the DISK or PRINT options, the specification of up to four additional filenames, and the specification of up to four TXTLIBs to be included in the GLOBAL TXTLIB command.

LOAD PROCESSING

This foreground option (option 4.6) provides a menu interface to the LOAD command (Figure 70). The menu allows a simple LOAD and GENMOD command to be issued with the specified options, or a user-specified EXEC to be issued for more complex load functions with multiple filenames, INCLUDEs, etc.



```
------------------------------ FOREGROUND LOAD ------------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX       ===> TEST      ===> MASTER     ===>
   TYPE    ===> TEXT
   MEMBER  ===> TOP_

CMS FILE:
   FILENAME ===>
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>

USER EXEC ===>               (USER'S EXEC TO BE RUN INSTEAD OF LOAD/GENMOD)
OPTIONS:
  LOAD    ===> CLEAR MAP
  GENMOD  ===> MAP

ADDITIONAL CONCATENATED TXTLIBS:
          ===>           ===>           ===>           ===>
```

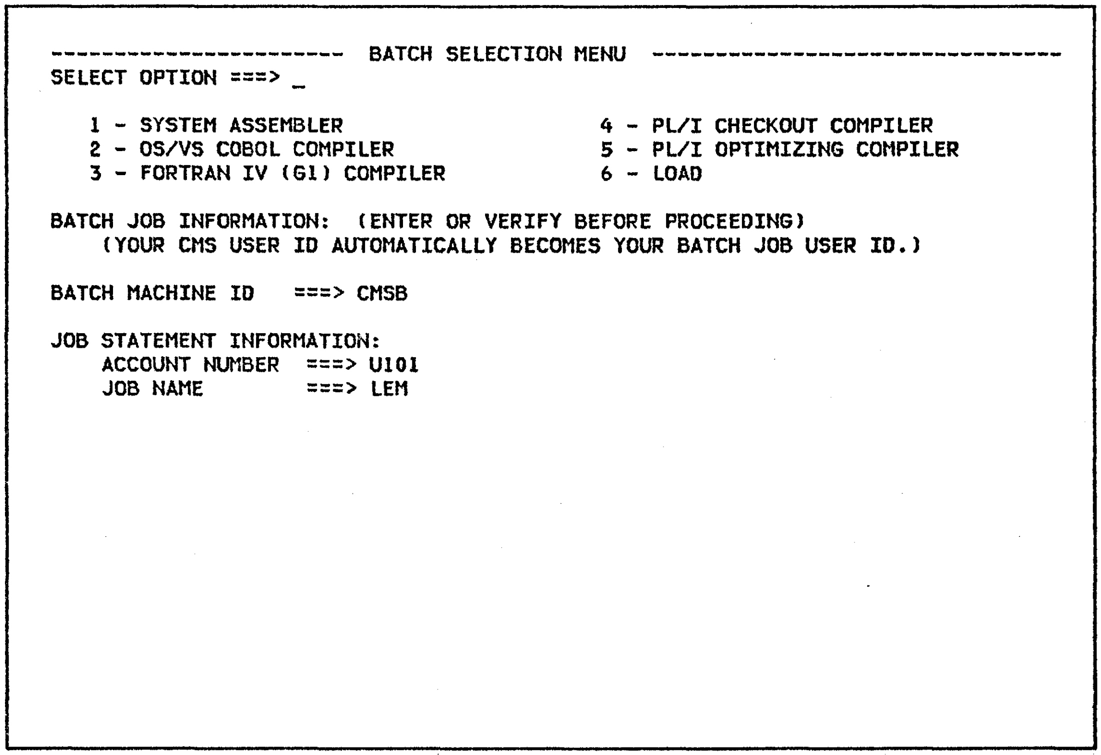
Figure 70. Foreground - LOAD menu

BATCH PROCESSING (OPTION 5)

The batch option provides an interface with standard language processors for compilation and assembly in the CMS batch machine of programs stored in SPF libraries or CMS files. It also provides a menu interface for batch execution of the CMS LOAD command.

When batch is selected (primary option 5), a menu is displayed to allow the user to select a language processor, and to enter and/or verify job statement parameters (Figure 71). When the user enters the desired information and presses the ENTER key, SPF displays the appropriate processor menu.

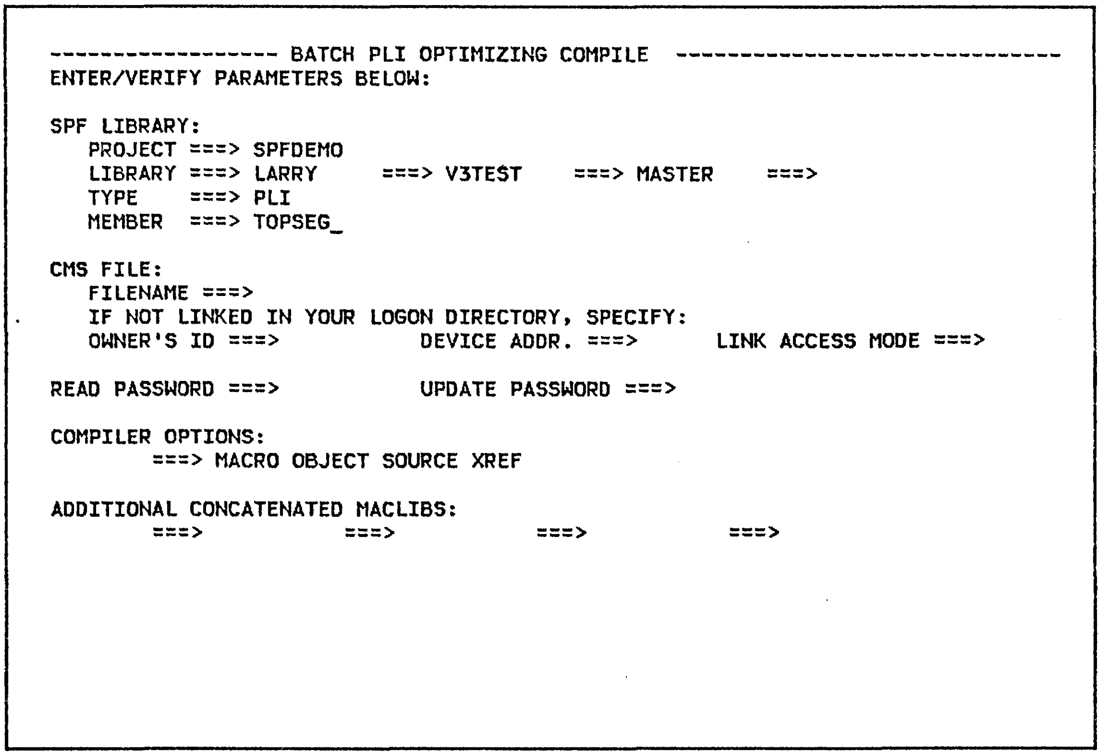
It is possible to bypass the secondary menu by entering two numbers, separated by a decimal point, on the primary option menu. For example, "5.2" to select a batch COBOL compile. Note, however, that use of this procedure does not allow verification or changes to the job statement parameters, nor does it allow the generation of multiple compilations (multiple job steps) within the same job.



```
----------------------- BATCH SELECTION MENU -------------------------------
SELECT OPTION ===> _

   1 - SYSTEM ASSEMBLER                       4 - PL/I CHECKOUT COMPILER
   2 - OS/VS COBOL COMPILER                   5 - PL/I OPTIMIZING COMPILER
   3 - FORTRAN IV (G1) COMPILER               6 - LOAD

BATCH JOB INFORMATION:  (ENTER OR VERIFY BEFORE PROCEEDING)
    (YOUR CMS USER ID AUTOMATICALLY BECOMES YOUR BATCH JOB USER ID.)

BATCH MACHINE ID    ===> CMSB

JOB STATEMENT INFORMATION:
    ACCOUNT NUMBER  ===> U101
    JOB NAME        ===> LEM
```

Figure 71. Batch selection menu

When job information has been entered and a language processor selected, a menu is displayed to allow the user to specify SPF library information or CMS filename, and processor options. Figure 72 shows an example for the PL/I optimizing compiler. The project name, first library name, type qualifier, and member name must be specified to compile a program whose source is in an SPF library. The filename must be specified to compile a CMS file with a filetype of PLI or PLIOPT. (If the CMS filename is entered, it overrides any SPF library specification.) The other parameters are optional, except as noted below.



```
------------------ BATCH PLI OPTIMIZING COMPILE -----------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> LARRY      ===> V3TEST    ===> MASTER     ===>
   TYPE    ===> PLI
   MEMBER  ===> TOPSEG_

CMS FILE:
   FILENAME ===>
   IF NOT LINKED IN YOUR LOGON DIRECTORY, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>           UPDATE PASSWORD ===>

COMPILER OPTIONS:
        ===> MACRO OBJECT SOURCE XREF

ADDITIONAL CONCATENATED MACLIBS:
        ===>           ===>           ===>           ===>
```

Figure 72. Batch - PL/I optimizing compiler example

When the user has filled in the menu entries and pressed the ENTER key, SPF generates the appropriate commands to be run in the batch machine. After SPF has generated the commands, it returns to the batch selection menu (unless that menu was bypassed by entering two numbers on the primary menu), with a message "job step generated" displayed in the short message area (line 1). If the batch selection menu was bypassed, SPF submits the generated commands and returns to the primary option menu. In either case, the user may exit from the processor menu without generating any commands by pressing the End PF key rather than the ENTER key.

When the batch menu is redisplayed, the job statement parameters are shown for information only -- they are no longer intensified and may not be overtyped. At this point, the user may select another processor to cause more commands (another job step) to be generated, enter CANCEL to return to the primary option menu without submitting the job, or press the End PF key (or Return PF key) to cause the generated commands to be submitted to the batch machine for execution.

If the job is submitted, a confirmation message is displayed, accompanied by a "HOLDING" message in the lower right-hand corner of the screen. When the user presses the PA2 key, SPF returns to the the primary option menu.

The batch machine sends messages to the user indicating the progress of the job. When the job has completed, the SPF reader utiltiy (option 3.7) may be used to retrieve TEXT and LISTING files produced by the batch job.

Caution: Use of the batch processing option in split screen mode should be avoided if the other logical screen is also being used to generate output for the virtual punch. The data will be intermixed on the punch if the user jumps back and forth between the screens. No data will be lost, but the results may be confusing.

The compiler menu shown in Figure 72 is typical of the batch language processor menus. In the figure, a library concatenation sequence of three libraries has been specified. The concatenation order is:

SPFDEMO.LARRY.PLI
SPFDEMO.V3TEST.PLI
SPFDEMO.MASTER.PLI

Before invoking the batch process, SPF will scan the concatenated sequence of libraries to find the member to be assembled or compiled (member TOPSEG in this example). A CMS file with a filetype acceptable to the language processor (PLI in this example) will be created on the user's A-disk by extracting the member from the first library in the sequence in which it occurs.

For any library in the sequence that is a collection of sequential files, a MACLIB will be created on the batch machine's A-disk to enable the use of COPY or INCLUDE statements. (For performance reasons, it is recommended that high-level libraries be specified as MACLIBs.)

Any temporary files created by SPF will be erased at the completion of the batch process.

Note the following points:

1. The concatenation sequence serves two purposes: It is used by SPF to locate the primary member to be compiled, and it allows inclusion of subsidiary members referenced by INCLUDE statements in the PL/I source.

2. Other MACLIBs can be appended to the GLOBAL statement, following the SPF libraries, by specifying them in the desired order in the "additional concatenated MACLIBs" fields.

3. The TEXT and LISTING files produced by the batch process are placed in the user's virtual reader using the DISK DUMP command. The SPF reader utility (option 3.7) may be used to retrieve these files following completion of the job.

4. If the source is specified as a CMS file (rather than an SPF library), the filetype must be whatever is acceptable to the language processor (PLI or PLIOPT in this example).

5. In specifying a CMS file that would require a LINK and ACCESS (i.e., if the file is on a minidisk not defined by an MDISK or LINK statement in the user's own LOGON directory), the user must provide the necessary LINK information: The user's id whose directory does contain that information, the virtual device address of the minidisk on which the source resides, and the desired link access mode (e.g, RR).

6. The language processor options are passed to the processor exactly as specified by the user. Up to 15 tokens may be specified as processor options. Only those options specified on the menu are passed to the language processor; SPF does not automatically generate any options.

The processor menus for Assembler and other compilers are similar to the one for the PL/I Optimizer. The meanings of the options differ among the several language processors. Check the user's guide for the specific language for an explanation of the processor options.

Compiler output is handled differently in the following cases:

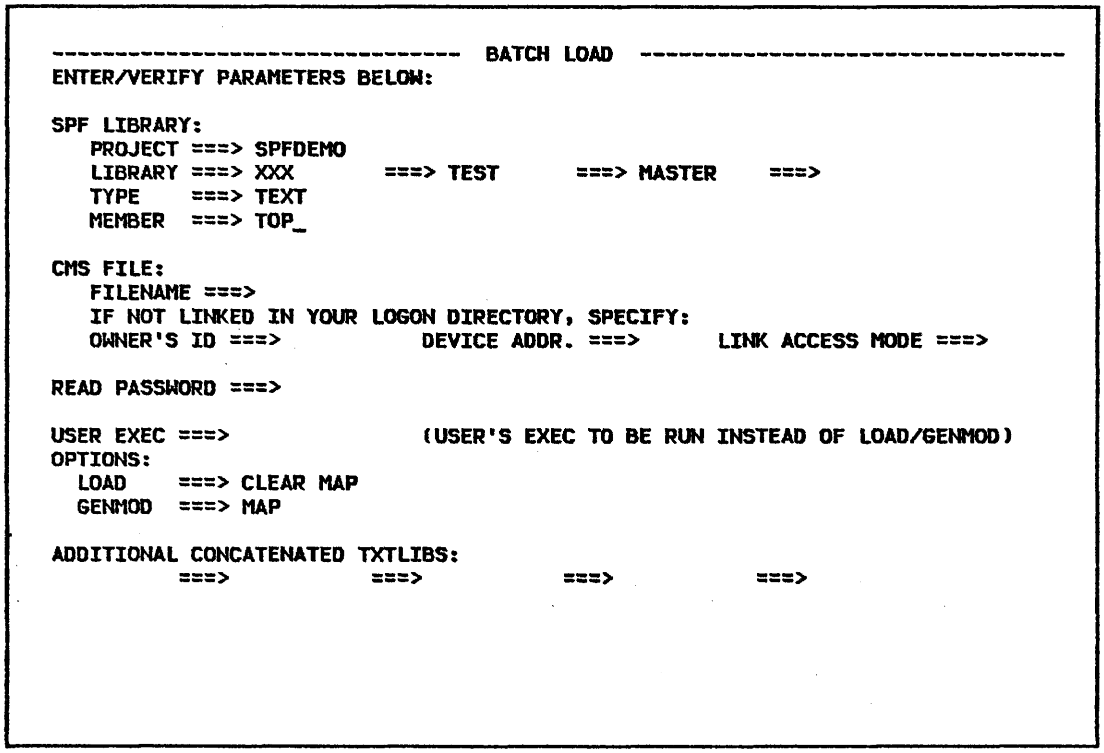
- When the TEST option is specified for the FORTRAN compiler, a special TEXT file is produced for later use by the FORTRAN interactive debug facility. The file is punched to the user's virtual reader, as is the generated FORTRAN file.

- When the TEST option is specified for the COBOL compiler, a SYSUT5 file is produced for later use by the COBOL interactive debug facility. The file is punched to the user's virtual reader.

- When the OBJECT option is specified for the PL/I Checkout compiler, a special TEXT file (a "link edit stub") and an ITXTLIB file (interpretable text) are produced for later use with the PLICR command. Both files are punched to the user's virtual reader.

LOAD PROCESSING

This batch option (option 5.6) provides a menu interface to the LOAD command (Figure 73). The menu allows a simple LOAD and GENMOD command to be issued with the specified options, or a user-specified EXEC to be issued for more complex load functions with multiple filenames, INCLUDEs, etc.

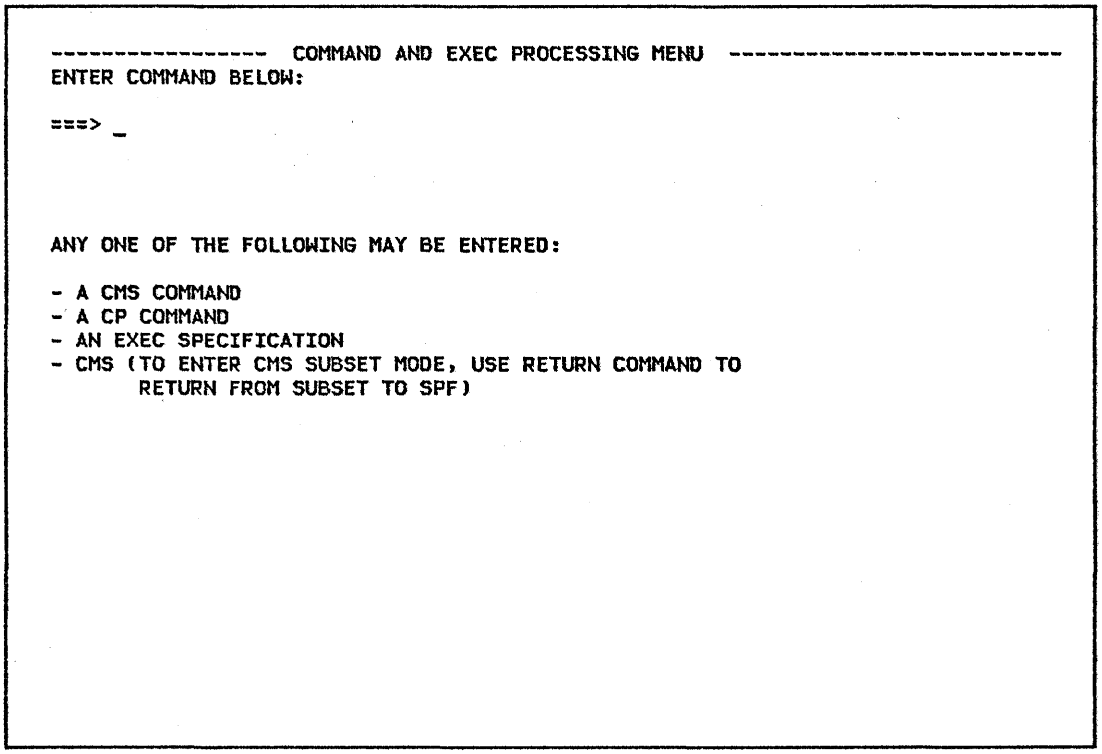


```
-------------------------------- BATCH LOAD --------------------------------
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX        ===> TEST       ===> MASTER     ===>
   TYPE    ===> TEXT
   MEMBER  ===> TOP_

CMS FILE:
   FILENAME ===>
   IF NOT LINKED IN YOUR LOGON DIRECTORY, SPECIFY:
   OWNER'S ID ===>           DEVICE ADDR. ===>      LINK ACCESS MODE ===>

READ PASSWORD ===>

USER EXEC ===>               (USER'S EXEC TO BE RUN INSTEAD OF LOAD/GENMOD)
OPTIONS:
  LOAD    ===> CLEAR MAP
  GENMOD  ===> MAP

ADDITIONAL CONCATENATED TXTLIBS:
          ===>           ===>           ===>           ===>
```

Figure 73. Batch - LOAD menu

CMS COMMAND (OPTION 6)

The CMS command option allows direct entry of a CMS command, CP command, EXEC, or entry to CMS subset under SPF. When this option is selected (primary option 6), the CMS command menu is displayed (Figure 74).



```
------------------ COMMAND AND EXEC PROCESSING MENU --------------------------
ENTER COMMAND BELOW:

===> _



ANY ONE OF THE FOLLOWING MAY BE ENTERED:

- A CMS COMMAND
- A CP COMMAND
- AN EXEC SPECIFICATION
- CMS (TO ENTER CMS SUBSET MODE, USE RETURN COMMAND TO
       RETURN FROM SUBSET TO SPF)
```

Figure 74. CMS command menu

The command or EXEC is entered to the right of the arrow, using standard CMS syntax. It may be continued on the next two lines, if necessary. Entering "CMS" causes entry to the CMS subset. The user exits from the CMS subset by issuing a RETURN command. While in subset mode, the PF keys and PA keys revert to standard CMS definitions.

After the CMS command has been typed, the user may press the ENTER key to cause execution of the command. Typically, the command and any resulting output will be displayed on the screen as if the command had been entered in a normal CMS environment. (The return code presentation may appear slightly different.) Following the last line displayed by the command, a "MORE..." message will be displayed in the lower right-hand corner of the screen. The user should then press PA2.

When the PA2 key is pressed, the menu is redisplayed. The command which was just executed is pre-entered to the right of the arrow. The user may then enter another command, or press the End PF key to return to the primary option menu.

Option 6 should not be used for entering SPF recursively, logging off, or issuing commands of a privileged class beyond the user's class. The results of any of the above are unpredictable.

TUTORIAL (OPTION 7)

The tutorial option provides immediate online reference and instruction on how to use SPF. It may be invoked from the primary option menu or via the Help PF key. The tutorial may be viewed sequentially from beginning to end, or it may be viewed randomly by selecting specific topics from an alphabetic index or table of contents.

When the tutorial is invoked from the primary option menu (option 7), introductory pages are displayed to explain how the tutorial works (Figure 75). Following the introduction, a table of contents is displayed from which the user may select a topic by entering the desired section number.

When the tutorial is invoked via the Help PF key, the appropriate section of the tutorial is entered based on what the user was doing when he pressed Help.

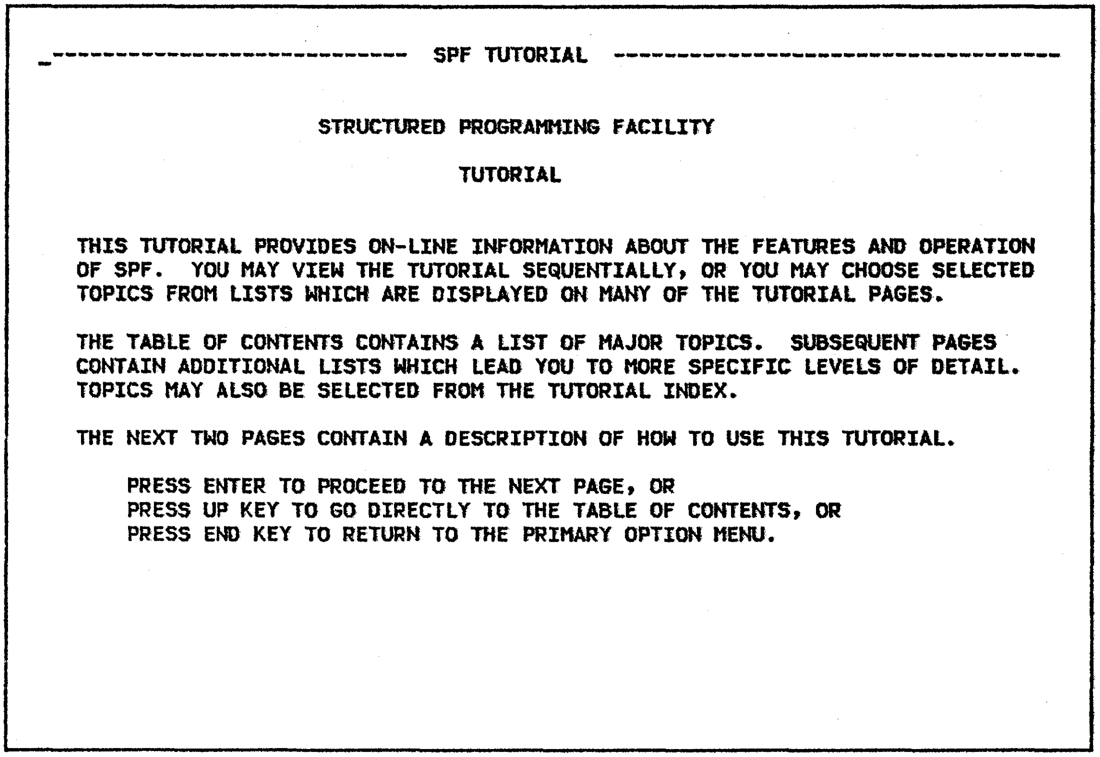
During execution of the tutorial, the four scroll PF keys are interpreted as follows:

UP	- display higher level list of topics
DOWN	- not used
LEFT (back)	- display previous tutorial page
RIGHT (next)	- display next tutorial page

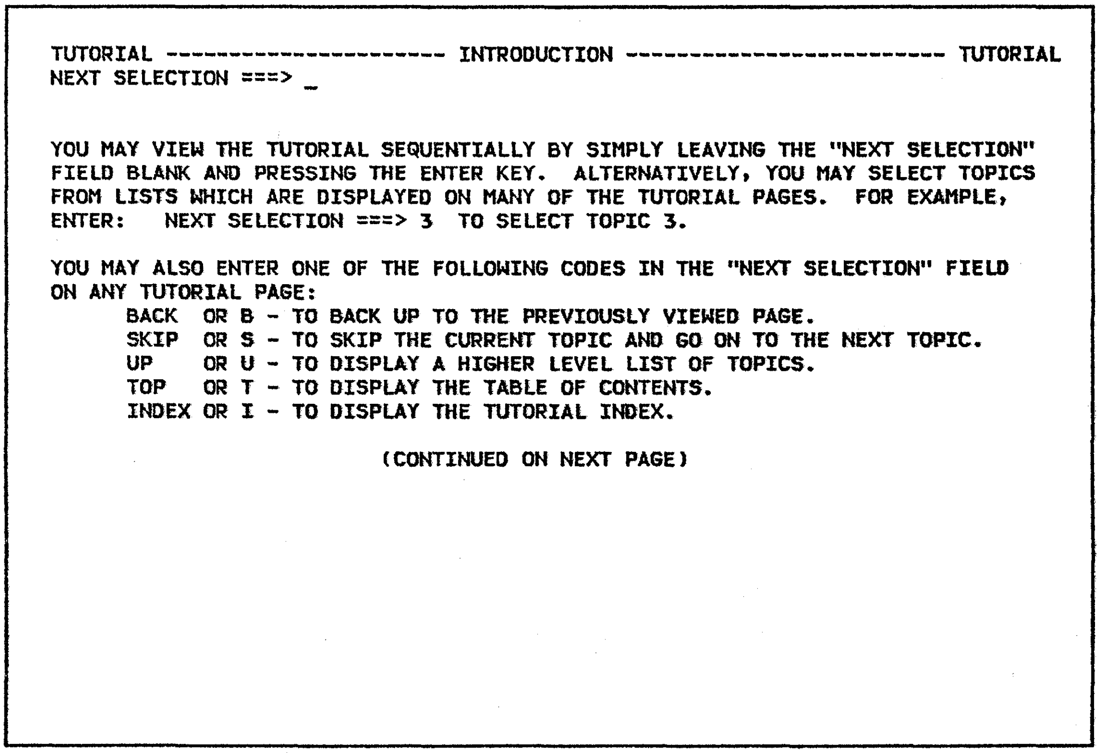
The Help PF key may be pressed at any time to display a one-page summary of how to use the tutorial.

The tutorial function is terminated by pressing the End PF key. This causes a return either to the primary option menu or to the display from which the user requested help.

Note: Each page of the tutorial is formatted to occupy 23 lines of the screen. If the tutorial is viewed in split-screen mode on a 24 line screen, the split line should be positioned either to the top or bottom line of the screen.



```
_---------------------------- SPF TUTORIAL -----------------------------------

                        STRUCTURED PROGRAMMING FACILITY

                                   TUTORIAL

  THIS TUTORIAL PROVIDES ON-LINE INFORMATION ABOUT THE FEATURES AND OPERATION
  OF SPF.  YOU MAY VIEW THE TUTORIAL SEQUENTIALLY, OR YOU MAY CHOOSE SELECTED
  TOPICS FROM LISTS WHICH ARE DISPLAYED ON MANY OF THE TUTORIAL PAGES.

  THE TABLE OF CONTENTS CONTAINS A LIST OF MAJOR TOPICS.  SUBSEQUENT PAGES
  CONTAIN ADDITIONAL LISTS WHICH LEAD YOU TO MORE SPECIFIC LEVELS OF DETAIL.
  TOPICS MAY ALSO BE SELECTED FROM THE TUTORIAL INDEX.

  THE NEXT TWO PAGES CONTAIN A DESCRIPTION OF HOW TO USE THIS TUTORIAL.

      PRESS ENTER TO PROCEED TO THE NEXT PAGE, OR
      PRESS UP KEY TO GO DIRECTLY TO THE TABLE OF CONTENTS, OR
      PRESS END KEY TO RETURN TO THE PRIMARY OPTION MENU.
```



```
TUTORIAL ----------------------- INTRODUCTION ------------------------- TUTORIAL
NEXT SELECTION ===> _


YOU MAY VIEW THE TUTORIAL SEQUENTIALLY BY SIMPLY LEAVING THE "NEXT SELECTION"
FIELD BLANK AND PRESSING THE ENTER KEY.  ALTERNATIVELY, YOU MAY SELECT TOPICS
FROM LISTS WHICH ARE DISPLAYED ON MANY OF THE TUTORIAL PAGES.  FOR EXAMPLE,
ENTER:   NEXT SELECTION ===> 3  TO SELECT TOPIC 3.

YOU MAY ALSO ENTER ONE OF THE FOLLOWING CODES IN THE "NEXT SELECTION" FIELD
ON ANY TUTORIAL PAGE:
      BACK  OR B - TO BACK UP TO THE PREVIOUSLY VIEWED PAGE.
      SKIP  OR S - TO SKIP THE CURRENT TOPIC AND GO ON TO THE NEXT TOPIC.
      UP    OR U - TO DISPLAY A HIGHER LEVEL LIST OF TOPICS.
      TOP   OR T - TO DISPLAY THE TABLE OF CONTENTS.
      INDEX OR I - TO DISPLAY THE TUTORIAL INDEX.

                            (CONTINUED ON NEXT PAGE)
```

Figure 75. Tutorial - first 2 pages

APPENDIX A: SAMPLE PROBLEM

This sample problem is a brief scenario of SPF terminal operations (Figure 76 through Figure 78). It is intended to serve as a demonstration that SPF has been properly installed and is operational. It may also be useful for new users as a quick introduction to SPF.

The scenario requires the installation of file SAMPLE ASSEMBLE (included on the SPF basic distribution tape), which should be available for read-only access on the SPF system mini-disk.

Prior to starting this sample problem, you should make sure that there are no files in your virtual reader.

ACTION	RESULT
1. LOGON AND IPL CMS (IF REQUIRED)	1. READY MESSAGE
2. TYPE "SPF"	2. SPF PRIMARY OPTION MENU
3. SELECT UTILITIES (OPTION 3)	3. UTILITY SELECTION MENU
4. SELECT FILE UTILITY (OPTION 2)	4. FILE UTILITY MENU
5. SELECT "SPECIFY" (S) AND ENTER: PROJECT : SAMPLE LIBRARY : USERID (YOUR VM USERID) TYPE : ASM	5. SPECIFY NEW LIBRARY MENU
6. ENTER THE FOLLOWING: ORGANIZATION : S RECORD FORMAT : F RECORD LENGTH : 80 LINK ACCESS MODE : W (LEAVE OTHER FIELDS BLANK)	6. 'SAMPLE.USERID.ASM' IS SPECIFIED AS AN SPF LIBRARY. FILE UTILITY MENU IS REDISPLAYED.
7. SELECT "SPECIFY" AGAIN AND ENTER: PROJECT : SAMPLE (SAME AS BEFORE) LIBRARY : USERID (SAME AS BEFORE) TYPE : TEXT	7. SPECIFY NEW LIBRARY MENU
8. (SAME ORGANIZATION, RECORD FORMAT, RECORD LENGTH, AND MODE LETTER SHOULD BE PRE-ENTERED). PRESS ENTER KEY.	8. 'SAMPLE.USERID.TEXT' IS SPECIFIED AS AN SPF LIBRARY. FILE UTILITY MENU IS REDISPLAYED.
9. PRESS PF3 (END KEY)	9. UTILITY SELECTION MENU
10. PRESS PF3	10. SPF PRIMARY OPTION MENU
11. SELECT MOVE/COPY (OPTION 3.3)	11. MOVE/COPY UTILITY MENU
12. SELECT COPY (C) AND ENTER: CMS FILE ID : SAMPLE ASSEMBLE	12. SECOND MOVE/COPY MENU
13. BLANK OUT CMS FILE ID AND ENTER: PROJECT : SAMPLE LIBRARY : USERID (YOUR VM USERID) TYPE : ASM MEMBER : ASMTEST	13. SAMPLE ASSEMBLE FILE IS COPIED TO YOUR LIBRARY (SAMPLE.USERID.ASM) AS MEMBER ASMTEST. THE FIRST MOVE/COPY MENU IS REDISPLAYED.
14. PRESS PF3	14. SPF PRIMARY OPTION MENU
15. SELECT BROWSE (OPTION 1)	15. BROWSE DATASET MENU
16. BLANK OUT CMS FILE ID AND ENTER: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : (LEAVE BLANK)	16. BROWSE MEMBER SELECTION LIST OF SPF LIBRARY 'SAMPLE.USERID.ASM'. ONLY ONE MEMBER (ASMTEST) IN LIST.
17. SELECT MEMBER ASMTEST (ENTER S AHEAD OF MEMBER NAME)	17. BROWSE DATA DISPLAY OF SAMPLE.USERID.ASM(ASMTEST)
18. SCROLL FORWARD ONE PAGE (PF8)	18. SECOND PAGE OF ASMTEST
19. SCROLL BACKWARD ONE PAGE (PF7)	19. FIRST PAGE OF ASMTEST AGAIN
20. ENTER COMMAND INPUT : ===> FIND COMMENT	20. STRING "COMMENT" IS INTENSIFIED.
21. FIND NEXT OCCURRENCE OF "COMMENT" BY PRESSING PF5	21. SECOND OCCURRENCE OF "COMMENT" IS INTENSIFIED.

Figure 76. Sample problem scenario (part 1 of 3)

ACTION	RESULT
22. PRESS PF3	22. MEMBER SELECTION LIST
23. PRESS PF3	23. BROWSE FILE MENU
24. PRESS PF3	24. SPF PRIMARY OPTION MENU
25. SELECT EDIT (OPTION 2)	25. EDIT FILE MENU
26. ENTER SPF LIBRARY: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ASMTEST	26. EDIT DATA DISPLAY OF SAMPLE.USERID.ASM(ASMTEST)
27. FIND THE LINE CONTAINING THE STRING "COMMENT" BY USING THE COMMAND ===> FIND COMMENT	27. CURSOR IS POSITIONED TO THE STRING AND SEQUENCE NUMBER IS INTENSIFIED.
28. DELETE THE COMMENT BY PRESSING ERASE EOF	28. THE RIGHT OF LINE IS ERASED
29. REPEAT THE NEXT COMMENT LINE BY OVERTYPING THE FIRST DIGIT OF THE LINE NUMBER WITH "R".	29. THE LINE IS REPEATED
30. ENTER THE COMMAND ===> PRINT ON	30. THIS SETS PRINT MODE ON FOR AUTOMATIC SOURCE LISTINGS.
31. TRY OUT MORE EDIT COMMANDS IF YOU LIKE, BUT REMEMBER THIS PROGRAM WILL BE ASSEMBLED LATER.	31. GOOD LUCK
32. PRESS PF3	32. MEMBER ASMTEST IS SAVED IN LIBRARY SAMPLE.USERID.ASM AND A LISTING OF THE MEMBER IS PLACED IN THE SPF LISTING FILE. THE EDIT FILE MENU IS DISPLAYED.
33. PRESS PF3	33. SPF PRIMARY OPTION MENU
34. SELECT FOREGROUND (OPTION 4)	34. FOREGROUND SELECTION MENU
35. SELECT SYSTEM ASSEMBLER	35. FOREGROUND ASSEMBLY MENU
36. ENTER ASSEMBLER INPUT: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ASMTEST ASSEMBLER OPTIONS : LIST TEST RENT	36. ASSEMBLER IS INVOKED. TERMINAL OUTPUT IS WRITTEN AT THE TOP OF A BLANK SCREEN. WHEN "MORE..." APPEARS, PRESS PA2 TO CONTINUE. TEXT FILE IS PLACED IN YOUR SPF LIBRARY SAMPLE.USERID.TEXT AS MEMBER ASMTEST, AND LISTING FILE (NAMED ASMTEST) IS WRITTEN ON YOUR A-DISK.
37. PRESS PF3	37. FOREGROUND SELECTION MENU
48. PRESS PF3	38. SPF PRIMARY OPTION MENU
39. SELECT BATCH (OPTION 5)	39. BATCH SELECTION MENU
40. SELECT ASSEMBLER (OPTION 1) AND ENTER JOB INFORMATION AS REQUIRED BY YOUR INSTALLATION.	40. BATCH ASSEMBLY MENU

Figure 77. Sample problem scenario (part 2 of 3)

<table>
<tr><th>ACTION</th><th>RESULT</th></tr>
<tr><td>41. ENTER ASSEMBLER INPUT:
PROJECT : SAMPLE
LIBRARY : USERID
TYPE : ASM
MEMBER : ASMTEST
ASSEMBLER
OPTIONS : TERM
OBJ NODECK LIST</td><td>41. CONTROL COMMANDS FOR ASSEMBLING ASMTEST ARE GENERATED. THE BATCH SELECTION MENU IS REDISPLAYED.</td></tr>
<tr><td>42. PRESS PF3</td><td>42. THE JOB IS SUBMITTED TO THE CMS BATCH MACHINE. WHEN "HOLDING" APPEARS, PRESS PA2. PRIMARY OPTION MENU IS REDISPLAYED.</td></tr>
<tr><td colspan="2">AS THE JOB EXECUTES IN THE BATCH MACHINE, STATUS MESSAGES WILL BE RETURNED TO YOU. IN PARTICULAR, YOU WILL RECEIVE A MESSAGE WHEN THE ASSEMBLY IS COMPLETE INDICATING THAT THE TEXT AND LISTING FILES HAVE BEEN SPOOLED TO YOUR READER. DO NOT EXECUTE STEP 44 OF THIS SCENARIO UNTIL YOU HAVE RECEIVED THE COMPLETION MESSAGE.</td></tr>
<tr><td>43. SELECT TUTORIAL (OPTION 7)</td><td>43. START OF SPF TUTORIAL IS DISPLAYED. FOLLOW THE DIRECTIONS TO LEARN MORE ABOUT SPF. WHEN YOU ARE DONE, PRESS PF3 TO RETURN TO THE PRIMARY OPTION MENU.</td></tr>
<tr><td>44. SELECT READER UTILITY (OPTION 3.7)</td><td>44. READER UTILITY MENU</td></tr>
<tr><td>45. SELECT INSERT (OPTION I) AND ENSURE THAT THE TARGET SPF LIBRARY FOR THE TEXT FILE IS SAMPLE.USERID.TEXT</td><td>45. READER UTILITY MENU IS REDISPLAYED, SHOWING THAT THE FILE WAS MOVED INTO YOUR SPF LIBRARY, AND THAT THE LISTING IS NOW THE NEXT FILE IN YOUR READER.</td></tr>
<tr><td>46. SELECT LOAD (OPTION L) AND ENTER:
RENAME TO : ASMTEST2 LISTING</td><td>46. READER UTILITY MENU IS REDISPLAYED, SHOWING THAT THE LISTING WAS LOADED ONTO YOUR A-DISK, AND THAT YOUR READER IS NOW EMPTY.</td></tr>
<tr><td>47. PRESS PF3</td><td>47. SPF PRIMARY OPTION MENU</td></tr>
<tr><td colspan="2">IF YOU HAVE FOLLOWED THIS SCENARIO, YOU SHOULD HAVE TWO LISTING FILES ON YOUR A-DISK, ONE NAMED ASMTEST (FROM THE FOREGROUND ASSEMBLY), AND THE OTHER NAMED ASMTEST2 (FROM THE BATCH ASSEMBLY). TRY BROWSE (OPTION 1) TO REVIEW THE LISTINGS, AND TRY EXPERIMENTING WITH OTHER SPF OPTIONS. WHEN YOU ARE DONE, RETURN TO THE PRIMARY OPTION MENU.</td></tr>
<tr><td>48. (AT PRIMARY OPTION MENU)
PRESS PF3</td><td>48. SPF TERMINATION MENU</td></tr>
<tr><td>49. SELECT OPTION P FOR BOTH THE LOG AND LIST FILES, AND PRESS ENTER.</td><td>49. THE LOG AND LIST FILES ARE CLOSED AND RELEASED FOR PRINTING.</td></tr>
<tr><td>50. YOU ARE NOW OUT OF SPF. TO EXIT FROM CMS TYPE "LOGOFF".</td><td>50. END OF SAMPLE</td></tr>
</table>

Figure 78. Sample problem scenario (part 3 of 3)

APPENDIX B: SPF LIBRARY SPECIFICATION AND USAGE

Before reading this appendix, please review the section entitled "SPF Libraries" on page 6 of this document.

GUIDELINES FOR LIBRARY SPECIFICATION

Each SPF library must be specified via the SPF file utility (option 3.2) before it can be used. The name of the library along with the following information must be specified:

SPF library attributes:	
Organization	(S = set of files, M = MACLIB, T = TXTLIB)
Record format	(F = fixed, V = variable)
Record length	(1 to 32767 bytes)
SPF library location:	
Owner's id	(blank for your user id)
Device addr.	(blank for '191')
Link access mode for update:	
Mode letters	(M, W, or MW)
For organization S (set of files):	
Filetype	(blank for SPF-generated name)
For organization M or T (MACLIB or TXTLIB):	
Filename	(blank for SPF-generated name)

Naming Conventions

An SPF library is always identified by a 3-component name (project name, library name, and type), where:

- "project name" is the common identifier for all libraries belonging to the same programming project.
- "library name" identifies the particular set of libraries (i.e., the level within the library hierarchy). For a private library, the user id is an appropriate library name.
- "type" identifies the type of information in the library. The SPF type qualifier need not conform to the standard CMS naming conventions, with one exception: If the library is to contain TEXT (object) modules, the SPF type qualifier must be TEXT regardless of whether the library organization is "S" (set of sequential files) or "T" (TXTLIB).

Each component of the name may be up to eight alphameric characters, of which the first must be alphabetic. The same rule applies to member names.

Library Organization

The library organization must be one of the following:

S - The library will be maintained as a set of CMS sequential files, all with the same filetype. The CMS filenames will be the same as the SPF member names, and the CMS filetype may be anything which uniquely identifies the set of files on the minidisk.

M - The library will be maintained as a CMS file with a filetype of MACLIB. The CMS filename may be anything which uniquely identifies the MACLIB on the minidisk. The member names in the MACLIB will be the same as the SPF member names.

T - The library will be maintained as a CMS file with a filetype of TXTLIB. The CMS filename may be anything which uniquely identifies the TXTLIB on the minidisk. The member names in the TXTLIB will be the same as the SPF member names.

Libraries containing documentation to be processed by the SCRIPT/VS utility (option 3.9) must have an organization of "S", but they need not have an SPF type qualifier of SCRIPT nor a CMS filetype of SCRIPT.

Libraries containing TEXT (object) modules must have an organization of "S" or "T", and the SPF type qualifier must be TEXT. (For organization "S" the CMS filetype need not be TEXT.)

Any other libraries may have an organization of "S" or "M". There are no restrictions on the SPF type qualifier, nor the CMS filetype (for organization "S").

Following are additional suggestions for specifying library organization:

- Libraries which are frequently updated (especially the lowest level libraries in a project hierarchy) should be assigned an organization of "S". This will eliminate the need for compresses.

- As a performance consideration, it may be desirable to assign higher level libraries an organization of "M" or "T" if one or more of the following applies:

  - The library has many members.

  - The library is infrequently updated.

  - The library contains members which are included or copied by other members during a compile or assembly. See "Foreground/Batch Processing", which follows, for a discussion of included members.

Record Format and Length

Libraries which have an organization of "M" or "T" must have a record format of "F" (fixed), and a record length of 80. Libraries which have an organization of "S" may have a record format of either "F" or "V" (variable), and a record length from 1 to 32767. Note, however, that the SPF editor will not process any data with a record length less than 10.

For format "F", the record length indicates the exact length of each record in the library. For format "V", the record length indicates the maximum allowable length of any record in the library.

Library Location

Each SPF library must be wholly contained on one minidisk, which is specified via owner's user id and virtual device address.

More than one SPF library may be assigned to the same minidisk, and SPF libraries may co-reside on the same minidisk with other CMS files (which are not SPF libraries). Be careful, however, to guard against conflicting CMS filetypes or filenames (see "Unique Identifier" and "Foreground/Batch Processing" which follow).

Following are additional suggestions for specifying library location:

- Typically, the lowest level libraries in a project hierarchy are private libraries and should reside on a minidisk which is owned by the user of the library (and should have an organization of "S").
- Higher level libraries in the project hierarchy are typically common libraries which are accessed for reading by anyone on the project, but are controlled (updated) by one designated individual. It is a good idea for these libraries to be on minidisks which are owned by the person who will be updating them, and which are write protected (i.e., no update password or an update password known only to the owner).
- If higher level libraries are to be protected against unauthorized reading by anyone outside the project, they should reside on minidisks which all have the same read password. The reason for this is that SPF menus provide a single read password which applies to all libraries in the concatenation sequence.

Link Access Mode

The link access mode for update must be one of the following:

W - Link will be successful only if no other user is currently accessing the minidisk (in either read or write mode).

M - Link will be successful if no other user is currently accessing the minidisk in write mode.

MW - Link will be successful in all cases.

If the library is to reside on a user's A-disk (which SPF assumes has a virtual device address of 191), do not specify "MW" for the link access mode. SPF will not protect against multiple write accessing to virtual device 191.

Note: As a general rule, multiple-write passwords should not be assigned to a user's A-disk. The absence of a such a password will prevent any attempt to link to the disk in "MW" mode.

For SPF libraries which are not on a user's A-disk, a link access mode of "MW" may be specified provided that all access to that minidisk (for SPF libraries or any other files) is through SPF. If some accesses to the minidisk will be done outside SPF, a link access mode of "M" or "W" should be specified.

SPF provides appropriate protection to guard against destructive contention in multiple read/write accessing of SPF libraries and CMS files, except as noted above for a user's A-disk.

Unique Identifier

The CMS filetype (for organization "S") or CMS filename (for organization "M" or "T") must uniquely identify which files on a minidisk belong to the SPF library. The recommended procedure is to let SPF assign a unique identifier by leaving the filetype and filename fields blank on the "specify" menu. The SPF-assigned identifier will be of the form:

SPFnnnnc

where "nnnn" is a four digit number, and "c" is S, M, or T.

Keep in mind the following points:

- For an SPF library, the CMS filetype (organization "S") or filename (organization "M" or "T") need not conform to any CMS naming convention.

- Since the SPF library can be referenced by project, library, and type on any menu, the CMS filetype or filename is generally transparent to the user.

If a filetype or filename is entered on the "specify" menu, it is the user's responsibility to ensure that the identifier is unique for the minidisk. If sequential files already exist on the minidisk with the specified filetype, or if MACLIBs or TXTLIBs already exist on the minidisk with the specified filename, the existing files will automatically be included in the SPF library. This may or may not be desired. In any case, the SPF file utility will display a warning on the "confirm specify" menu that existing files will be included in the SPF library.

PROCESSING CONSIDERATIONS

SPF maintains information on all SPF libraries specified to the VM system. Whenever a user enters project, library, and type on a menu, SPF determines the minidisk on which the library is located (user id and virtual device address), and then performs the necessary LINK and ACCESS commands on behalf of the user.

For the LINK command, SPF uses a link access mode of "RR" (for a read-only function such as browse), or whatever mode was entered when the library was specified (for an update function such as edit). The link access mode of "RR" alows the LINK to be successful even if another user currently has the disk accessed in read or write status.

For the ACCESS command, SPF selects the first available mode letter (in the collating sequence) which is not currently in use by the user who is accessing the library.

The minidisk containing the SPF library is automatically detached (and the mode letter is released) when the user finishes the selected SPF function.

Under certain circumstances, the automatic LINK and ACCESS can cause a temporary alteration to the user's normal search sequence for CMS files. Example:

> Suppose the user has explicitly accessed two minidisks, using mode letters A and C. Subsequently, the user specifies an SPF library on some menu (such as browse). SPF selects mode letter B to access the library. The user then enters split screen mode (without leaving browse), and specifies a CMS filename and filetype on another menu. If the specified file exists on both the B and C disks (but not on A), the B copy will be found first. The user may not be expecting this, since he did not explicitly access the B-disk.

<u>Foreground/Batch Processing</u>

The SPF foreground and batch processing options (options 4 and 5) automatically perform the following operations if an SPF library (or concatenated sequence of libraries) is specified:

- The library (or sequence of libraries) is scanned to find the specified member, and the member is copied to a temporary file, if required, with a CMS filetype which is acceptable to the particular processing program (e.g., ASSEMBLE, COBOL, PLI). The CMS filename for the temporary file will be the same as the SPF member name.

- For any libraries in the SPF concatenation sequence (including the first, or only, library) which are not organization "M", temporary MACLIBs are created and all members are copied from each SPF library to the corresponding MACLIB. A GLOBAL command is then executed to set up the MACLIB concatenation sequence. This allows members to be included (via INCLUDE or COPY statements) from the concatenated sequence of libraries during the compilation or assembly.

- For foreground processing, the temporary file and temporary MACLIBs will reside on the the user's A-disk. For batch, they will reside on the batch machine's A-disk. In either case, the temporary files are erased once the compilation or assembly has completed.

- The TEXT (object) module produced by the compilation or assembly is placed in an SPF library with the following name:

  Project - same as project name of source library
  Library - same as first library in concatenation sequence
  Type - TEXT

  For foreground processing, the TEXT module is automatically placed in the SPF TEXT library. For batch processing, the user may invoke the SPF reader utility (option 3.7) to cause the TEXT module to be inserted into the SPF TEXT library.

SPF Library Statistics

Whenever an SPF library is specified, a separate file is created on the same minidisk to contain the library statistics. This is a sequential file with a CMS filetype of SPFSTATS and a filename which is the same as the unique identifier of the SPF library.

Generally, the SPFSTATS file is transparent to the user, since the statistics are automatically created and maintained by the SPF editor (if STATS mode is on), and automatically displayed on member selection lists and printed on "index" listings.

Users should not attempt to edit or otherwise modify the SPFSTATS files. If an SPFSTATS file is modified or destroyed, however, only the statistics will be lost; the contents of the SPF library members will still be intact.

If members of the SPF library are modified outside of SPF (via some other editor) or are specified on an SPF menu via CMS file id (rather than project, library, and type), the statistics will not be properly updated, but the information in the member will still be acceptable to SPF.

Note: If a user deletes all lines in a member of an SPF library (using the SPF editor) and then saves the member, the statistics will show that the member still exists but has a length of zero. However, if the library is specified for a foreground or batch compilation or assembly, the member will not be found by the processing program. The proper way to delete a member (including its statistics) is to use the library utility (option 3.1).

APPENDIX C: SPF LISTING FORMATS

Under user option, the SPF editor will automatically generate a source listing upon completion of editing (if PRINT mode is on). Source listings may also be obtained via the "print" option of the SPF library utility (option 3.1) and the move/copy utility (option 3.3).

A sample source listing is shown in Figure 79.

For an SPF library, information at the top of the page includes project, library, type, member, current version, and modification level. The member name is also repeated at the right for ease of filing. For a CMS file, the file id (and member name for a MACLIB or TXTLIB) are indicated. The header lines also include user id, date and time that the listing was produced, and page number.

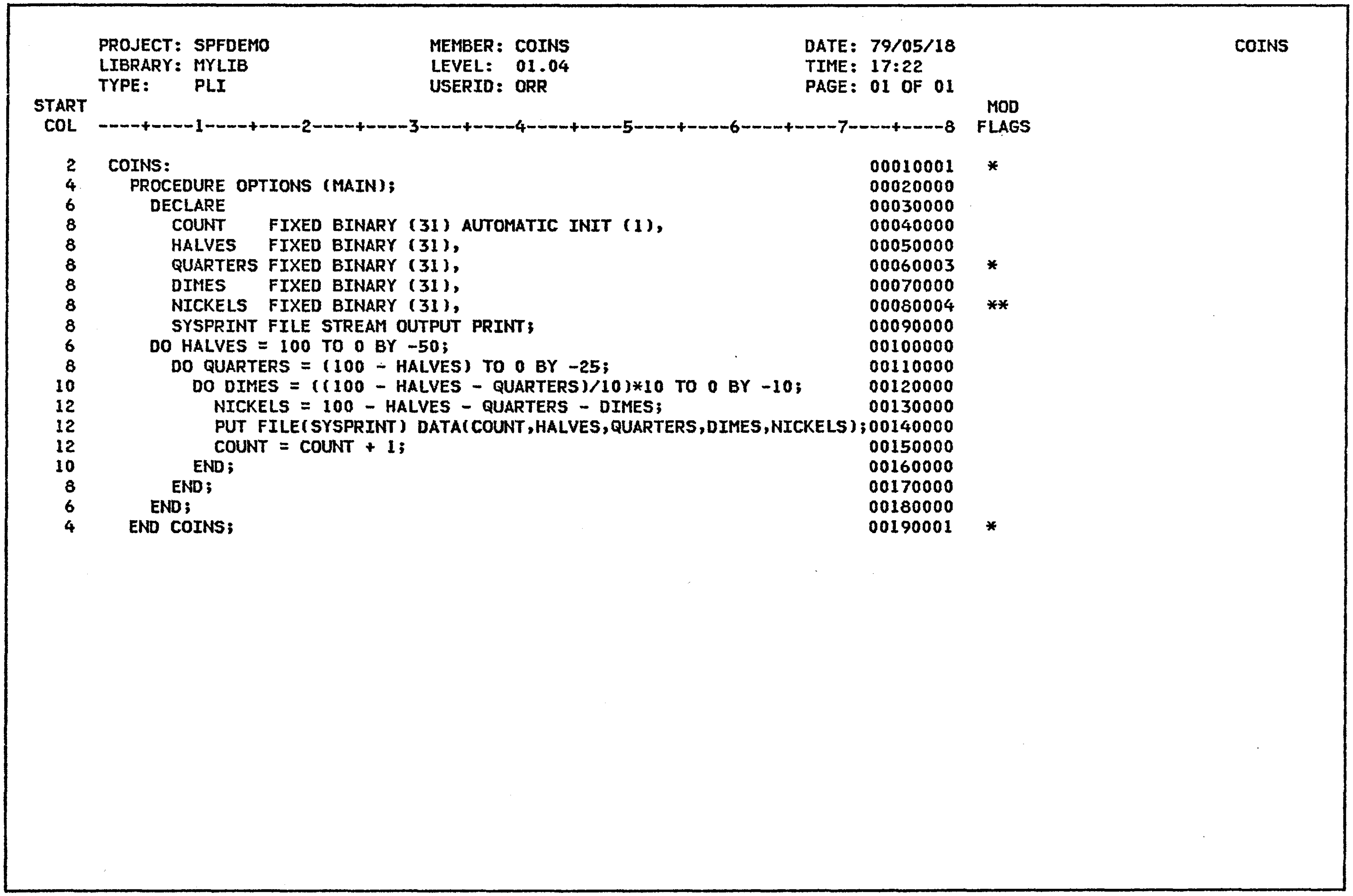
A column positioning line is printed following the heading and preceding the actual data. The "start column" is printed to the left of each line, indicating the position of the first non-blank character in each line. For SPF library members with statistics, asterisks may be printed to the right of each line, as follows:

- If the modification flag (columns 79-80) in the line is 00, no asterisks are printed.
- If the modification flag is non-zero but differs from the current modification level of the member, a single asterisk (*) is printed.
- If the modification flag is non-zero and has the same value as the current modification level of the member, two asterisks (**) are printed.

The asterisks allow the listing to be scanned quickly for lines that were added or changed since the version was created (*) and for lines that were added or changed during the last update (**).

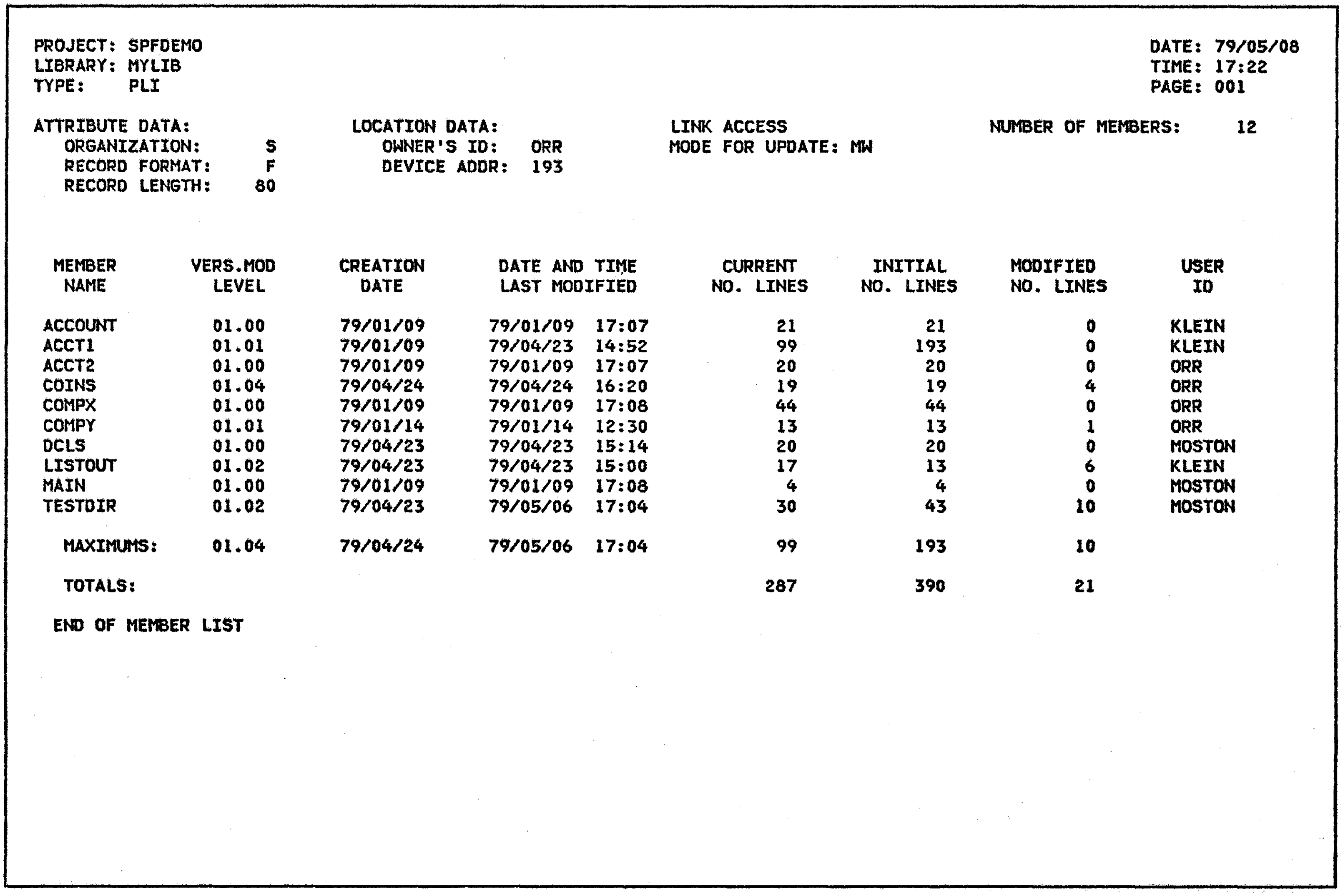
SPF provides index listings, upon user request, via the "index" option of the library utility (option 3.1). A sample index listing for an SPF source library is shown in Figure 80. The heading information includes: project, library, and type, date and time that the listing was produced, and page number. This is followed by general information about the library. Following this, the member name and statistics are printed for each member, arranged in alphabetical order.

Finally, a sample SPF log listing is shown in Figure 81. The log contains a message for each significant user action, such as saving edited data, moving members from one library to another, or submitting a batch job.



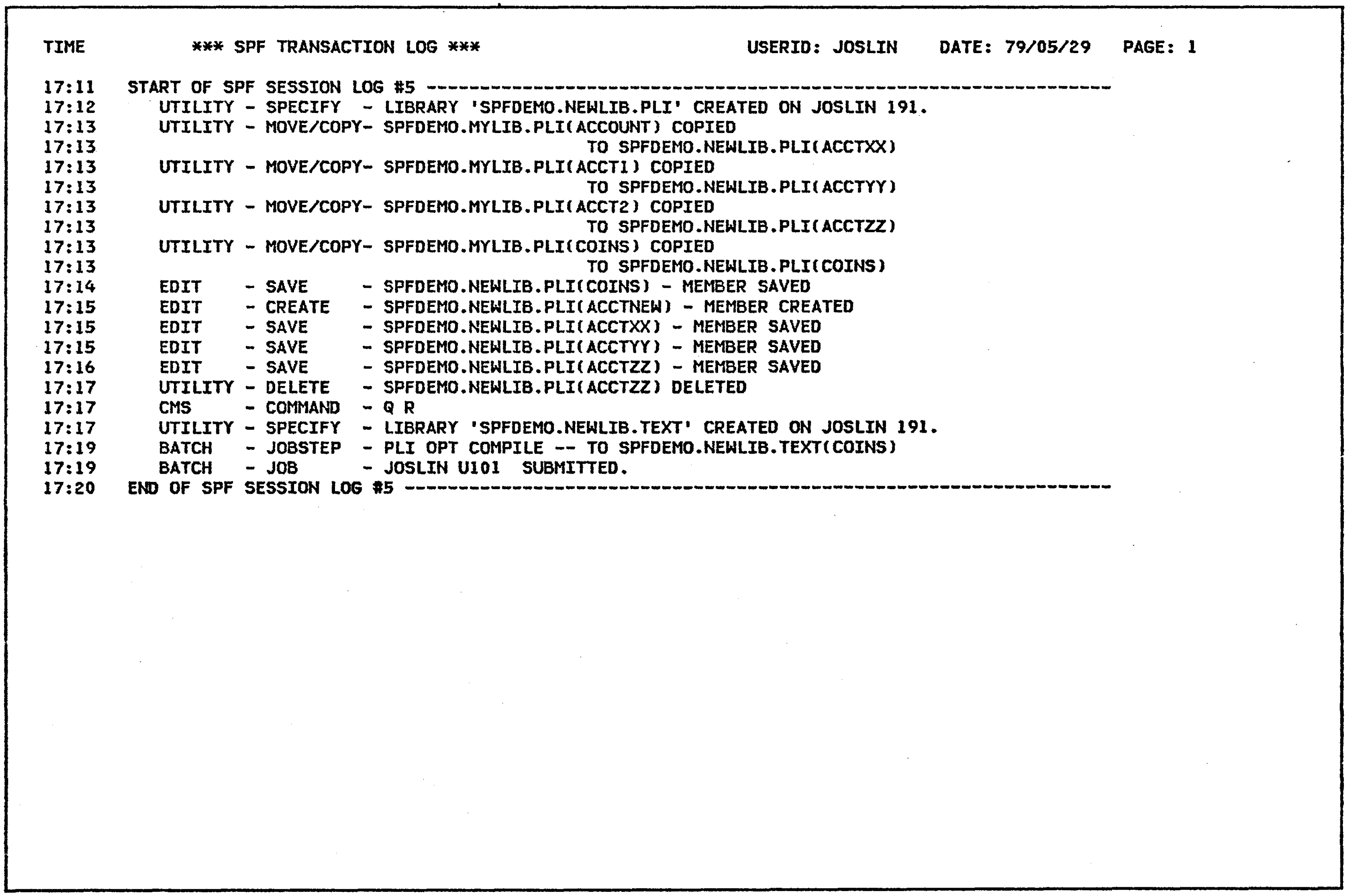
```
      PROJECT: SPFDEMO               MEMBER: COINS                      DATE: 79/05/18                          COINS
      LIBRARY: MYLIB                 LEVEL:  01.04                      TIME: 17:22
      TYPE:    PLI                   USERID: ORR                        PAGE: 01 OF 01
START                                                                                    MOD
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8  FLAGS

   2   COINS:                                                                 00010001   *
   4     PROCEDURE OPTIONS (MAIN);                                            00020000
   6       DECLARE                                                            00030000
   8         COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),                   00040000
   8         HALVES   FIXED BINARY (31),                                      00050000
   8         QUARTERS FIXED BINARY (31),                                      00060003   *
   8         DIMES    FIXED BINARY (31),                                      00070000
   8         NICKELS  FIXED BINARY (31),                                      00080004   **
   8         SYSPRINT FILE STREAM OUTPUT PRINT;                               00090000
   6       DO HALVES = 100 TO 0 BY -50;                                       00100000
   8         DO QUARTERS = (100 - HALVES) TO 0 BY -25;                        00110000
  10           DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;      00120000
  12             NICKELS = 100 - HALVES - QUARTERS - DIMES;                   00130000
  12             PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);00140000
  12             COUNT = COUNT + 1;                                           00150000
  10           END;                                                           00160000
   8         END;                                                             00170000
   6       END;                                                               00180000
   4     END COINS;                                                           00190001   *
```

Figure 79. Sample source listing
]



```
PROJECT: SPFDEMO                                                                                  DATE: 79/05/08
LIBRARY: MYLIB                                                                                    TIME: 17:22
TYPE:    PLI                                                                                      PAGE: 001

ATTRIBUTE DATA:               LOCATION DATA:               LINK ACCESS                 NUMBER OF MEMBERS:     12
   ORGANIZATION:      S          OWNER'S ID:   ORR         MODE FOR UPDATE: MW
   RECORD FORMAT:     F          DEVICE ADDR:  193
   RECORD LENGTH:    80


  MEMBER       VERS.MOD     CREATION      DATE AND TIME       CURRENT      INITIAL      MODIFIED      USER
   NAME          LEVEL        DATE        LAST MODIFIED      NO. LINES    NO. LINES    NO. LINES       ID

 ACCOUNT        01.00       79/01/09    79/01/09  17:07          21           21            0       KLEIN
 ACCT1          01.01       79/01/09    79/04/23  14:52          99          193            0       KLEIN
 ACCT2          01.00       79/01/09    79/01/09  17:07          20           20            0       ORR
 COINS          01.04       79/04/24    79/04/24  16:20          19           19            4       ORR
 COMPX          01.00       79/01/09    79/01/09  17:08          44           44            0       ORR
 COMPY          01.01       79/01/14    79/01/14  12:30          13           13            1       ORR
 DCLS           01.00       79/04/23    79/04/23  15:14          20           20            0       MOSTON
 LISTOUT        01.02       79/04/23    79/04/23  15:00          17           13            6       KLEIN
 MAIN           01.00       79/01/09    79/01/09  17:08           4            4            0       MOSTON
 TESTDIR        01.02       79/04/23    79/05/06  17:04          30           43           10       MOSTON

   MAXIMUMS:    01.04       79/04/24    79/05/06  17:04          99          193           10

   TOTALS:                                                      287          390           21

  END OF MEMBER LIST
```

Figure 80. Sample index listing - source library



```
TIME          *** SPF TRANSACTION LOG ***                         USERID: JOSLIN    DATE: 79/05/29   PAGE: 1

17:11   START OF SPF SESSION LOG #5 ----------------------------------------------------------------
17:12      UTILITY - SPECIFY  - LIBRARY 'SPFDEMO.NEWLIB.PLI' CREATED ON JOSLIN 191.
17:13      UTILITY - MOVE/COPY- SPFDEMO.MYLIB.PLI(ACCOUNT) COPIED
17:13                                              TO SPFDEMO.NEWLIB.PLI(ACCTXX)
17:13      UTILITY - MOVE/COPY- SPFDEMO.MYLIB.PLI(ACCT1) COPIED
17:13                                              TO SPFDEMO.NEWLIB.PLI(ACCTYY)
17:13      UTILITY - MOVE/COPY- SPFDEMO.MYLIB.PLI(ACCT2) COPIED
17:13                                              TO SPFDEMO.NEWLIB.PLI(ACCTZZ)
17:13      UTILITY - MOVE/COPY- SPFDEMO.MYLIB.PLI(COINS) COPIED
17:13                                              TO SPFDEMO.NEWLIB.PLI(COINS)
17:14      EDIT    - SAVE     - SPFDEMO.NEWLIB.PLI(COINS) - MEMBER SAVED
17:15      EDIT    - CREATE   - SPFDEMO.NEWLIB.PLI(ACCTNEW) - MEMBER CREATED
17:15      EDIT    - SAVE     - SPFDEMO.NEWLIB.PLI(ACCTXX) - MEMBER SAVED
17:15      EDIT    - SAVE     - SPFDEMO.NEWLIB.PLI(ACCTYY) - MEMBER SAVED
17:16      EDIT    - SAVE     - SPFDEMO.NEWLIB.PLI(ACCTZZ) - MEMBER SAVED
17:17      UTILITY - DELETE   - SPFDEMO.NEWLIB.PLI(ACCTZZ) DELETED
17:17      CMS     - COMMAND  - Q R
17:17      UTILITY - SPECIFY  - LIBRARY 'SPFDEMO.NEWLIB.TEXT' CREATED ON JOSLIN 191.
17:19      BATCH   - JOBSTEP  - PLI OPT COMPILE -- TO SPFDEMO.NEWLIB.TEXT(COINS)
17:19      BATCH   - JOB      - JOSLIN U101  SUBMITTED.
17:20   END OF SPF SESSION LOG #5 ------------------------------------------------------------------
```

Figure 81. Sample log listing

APPENDIX D: SUMMARY OF SPF COMMANDS

MEMBER LIST COMMANDS

The following commands may be entered in the command input field (line 2 of the logical screen) on member list displays.

LOCATE Causes direct scrolling to the specified member name (may be entered on any member list display).

SELECT Causes the specified member name to be selected for browse, edit, or SCRIPT/VS utility (invalid on other utility member list displays). Allows selection of a member which is not in the list.

On member list displays, the following one-character commands may be entered at the beginning of a line, ahead of the member name.

S (select) Causes the member to be selected for browse, edit, move/copy, reset, or SCRIPT/VS utility (invalid under the library utility).

P (print) Causes the member to be printed (valid only under the library utility).

R (rename) Causes the member to be renamed (valid only under the library utility). The new name must be entered to the right of the member name.

D (delete) Causes the member to be deleted (valid only under the library utility).

B (browse) Causes the member to be browsed (valid only under the library utility).

BROWSE COMMANDS

The following commands may be entered in the command input field (line 2 of the logical screen) under browse, and also under the SCRIPT/VS utility and the "display" option of the SPF project utility.

LOCATE Causes direct scrolling to the specified relative line number or label.

.xxxxx Establishes a label (SPF internal symbol) which is equated to the top line on the screen. It can then be used with the LOCATE command to cause scrolling directly to that line.

COLS Causes display of a line that identifies column positions.

RESET Removes the column identification line from display.

FIND Causes the data to be scanned for one or all occurrences of a character string. The cursor is repositioned to the beginning of the first occurrence, if it is found.

CAPS Causes character strings, entered as part of the FIND command, to be converted to upper case. CAPS OFF causes character strings to be left as-is.

HEX Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.

EDIT LINE COMMANDS

Under edit, line commands may be entered at the beginning of a line by overtyping the line number.

COLS Causes display of a line that identifies column positions.

I (insert) Causes a new line to be inserted after this line.

D (delete) Causes the line to be deleted.

R (repeat) Causes the line to be repeated and the duplicate copy to be inserted after this line.

M (move) Identifies a line to be moved.

C (copy) Identifies a line to be copied.

A (after) Identifies the line after which copied or moved lines are to be inserted.

B (before) Identifies the line before which copied or moved lines are to be inserted.

MASK Causes display of the insert mask, to allow pre-entry of data on inserted lines.

< (left) Causes the contents of the line to be "data" shifted left.

> (right) Causes the contents of the line to be "data" shifted right.

((left) Causes the contents of the line to be "column" shifted left.

) (right) Causes the contents of the line to be "column" shifted right.

BOUNDS Causes display of the column boundary line, to allow redefinition of boundary positions.

X (exclude) Causes the line to be excluded from display. May also be used to limit the scope of the FIND and CHANGE commands.

S (show) Causes an excluded line to be redisplayed.

F (first) Causes the first line in a block of excluded lines to be redisplayed.

L (last) Causes the last line in a block of excluded lines to be redisplayed.

TABS Causes display of the tab line, to allow definition of hardware, software, and logical tab positions.

TE Facilitates bulk text entry by allowing the user to type continuously without regard to the line number area.

TS Causes splitting of a line into two lines at the designated cursor position to facilitate the insertion of new material in existing text.

TF Causes reflowing of a paragraph of text from the cursor position to the end of the paragraph.

O (overlay) Causes data to be copied or moved "over" the existing data on one or more lines. Blank characters in the receiving line(s) are overlayed with corresponding characters in the source line(s).

The following line commands may be entered with double characters to indicate the beginning and end of a block of lines.

DD	- Block delete	<<	- Block "data" shift left
RR	- Block repeat	>>	- Block "data" shift right
MM	- Block move	((	- Block "column" shift left
CC	- Block copy	))	- Block "column" shift right
XX	- Block exclude	OO	- Block overlay

The following line commands may incorporate a number (n) consisting of one or more digits.

In	- Insert n lines following this line
Dn	- Delete n lines starting at this line
Rn	- Repeat this line n times
RRn	- Repeat block of lines n times
Mn	- Move n lines starting at this line
Cn	- Copy n lines starting at this line
An	- Insert n copies of the moved/copied line(s) after this line
Bn	- Insert n copies of the moved/copied line(s) before this line
<n	- Data-shift line left n positions
<<n	- Data-shift block of lines left n positions
>n	- Data-shift line right n positions
>>n	- Data-shift block of lines right n positions
(n	- Column-shift line left n positions
((n	- Column-shift block of lines left n positions
)n	- Column-shift line right n positions
))n	- Column-shift block of lines right n positions
Xn	- Exclude n lines starting at this line
Sn	- Show n lines in a block of excluded lines
Fn	- Show first n lines in a block of excluded lines
Ln	- Show last n lines in a block of excluded lines
TEn	- Insert n text entry lines following this line
TSn	- Insert n new lines between the split lines
TFn	- Flow text from left bound to column position n
On	- Overlay n lines

If a number is not entered, the default is 1 except for:

- The shift commands (which default to 2 column positions)

- The TE command (defaults to the number of lines remaining on the screen)

- The TF command (defaults to the specified column boundaries)

EDIT PRIMARY COMMANDS

Under edit, the following commands may be entered in the command input field (line 2 of the logical screen).

LOCATE Causes direct scrolling to the specified line number.

NUMBER Causes sequence numbers to be generated for any new lines that are created via insert, repeat, or copy. NUMBER OFF causes lines to be generated without numbers.

RENUM Renumbers each line and turns on NUMBER mode.

UNNUM Causes sequence numbers to be set to blanks and turns off NUMBER mode.

AUTONUM If edit is in NUMBER mode, causes sequence numbers to be automatically renumbered whenever a save is done. AUTONUM OFF bypasses automatic renumbering at save time.

STATS Causes SPF statistics to be generated or updated whenever a member of an SPF library is saved, created, or replaced. STATS OFF causes members to be stored without statistics.

PRINT Causes a source listing of the edited data to be automatically recorded in the SPF list file whenever data is saved by pressing the End PF key. PRINT OFF bypasses automatic source listings.

RECOVERY Enables the SPF edit recovery function. If there is a subsequent system failure, the user can recover the SPF edit session up to the point of failure. RECOVERY OFF disables edit recovery.

RESET Causes a general resetting of intensified messages, incomplete line commands, and special lines.

SUBMIT Causes the data being edited to be submitted to the job stream of another VM machine for execution.

SAVE Causes the data to be stored back into the edit file and editing to continue.

CANCEL Causes editing to be terminated without saving the data.

FIND Causes the data to be scanned for one or all occurrences of a character string. The cursor is repositioned to the beginning of the first occurrence, if it is found.

CHANGE Causes one or all occurrences of a character string to be replaced with a second string. The cursor is repositioned to the end of the first occurrence, if it is found.

COPY Causes data to be copied from another source and inserted at a point indicated with the A (after) or B (before) line command. The other source can be a member of an SPF library or MACLIB, or a sequential CMS file.

MOVE Same as COPY, except that the other source is deleted after the copy operation.

CREATE Causes a line or block of lines to be stored in any SPF library or MACLIB as a new member. The line(s) to be stored are designated with the C or CC (copy), or M or MM (move) line commands.

REPLACE Same as CREATE, except that it allows an existing member or a CMS sequential file to be replaced.

NULLS Replaces trailing blanks on the screen with null characters. NULL OFF causes trailing blanks to be sent to the screen as blanks.

TABS Enables the use of "hardware" or "logical" tabs at user defined positions (specified via the tabs line command). TABS OFF disables the use of "hardware" or "logical" tabs.

PROFILE Causes the display of the current setting of edit modes (NUMBER, AUTONUM, STATS, PRINT, CAPS, NULLS, TABS, HEX, and RECOVERY). Also displays the MASK, TABS, and BOUNDS lines whenever they have settings other than their defaults.

HEX Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.

CAPS Causes alphabetic data entered from the terminal to be translated to upper case, including FIND and CHANGE strings. CAPS OFF causes alphabetic data to be left as-is.

APPENDIX E: SPF COMMAND SYNTAX - QUICK REFERENCE SUMMARY

MEMBER LIST COMMANDS

Primary Commands:

```
LOCATE   member-name

SELECT   member-name
```

Line Selection Commands:

```
S (select) - except opt. 3.1

P (print)  - option 3.1 only
R (rename) - option 3.1 only
D (delete) - option 3.1 only
B (browse) - option 3.1 only
```

BROWSE COMMANDS

```
LOCATE  line-number/label

.xxxxx

COLS

RESET

FIND  string  [NEXT/ALL/FIRST/LAST/PREV]
              [CHARS/PREFIX/SUFFIX/WORD]
              [col-1 [col-2]]

CAPS  [ON/OFF]

HEX   [ON/OFF]  [VERT/DATA]
```

EDIT LINE COMMANDS

Basic Commands:

```
COLS
I, In               (insert)
D, Dn, DD           (delete)
R, Rn, RR, RRn      (repeat)
M, Mn, MM           (move)
C, Cn, CC           (copy)
A, An               (after)
B, Bn               (before)
```

Advanced Features:

```
MASK
<, <n, <<, <<n  (data left)
>, >n, >>, >>n  (data right)
(, (n, ((, ((n  (cols left)
), )n, )), ))n  (cols right)
BOUNDS
X, Xn, XX    (exclude)
S, Sn        (show)
F, Fn        (first)
L, Ln        (last)
TABS
```

Text Preparation Features:

```
TE, TEn     (text entry)
TS, TSn     (text split)
TF, TFn     (text flow)
O, On, OO   (overlay)
```

EDIT PRIMARY COMMANDS

General Commands:

```
LOCATE  line-number

NUMBER  [ON/OFF] [STD] [COBOL]

RENUM   [STD]  [COBOL]

UNNUM

AUTONUM  [ON/OFF]

STATS  [ON/OFF]

PRINT  [ON/OFF]

RECOVERY  [ON/OFF]

RESET

SUBMIT  machine-id

SAVE

CANCEL
```

Advanced Features:

```
COPY  [member-name]
MOVE  [member-name]

CREATE  [member-name]
REPLACE [member-name]

NULLS   [ON/OFF] [ALL]

TABS    [ON/OFF]
        [tab-char] [ALL]

PROFILE  [name] [number]

HEX  [ON/OFF]
     [VERT/DATA]
```

Text Preparation Features:

```
CAPS  [ON/OFF]
```

Find and Change Commands:

```
FIND  string-1  [NEXT/ALL/FIRST/LAST/PREV]
                [CHARS/PREFIX/SUFFIX/WORD]
                [X/NX]
                [col-1 [col-2]]

CHANGE  string-1  string-2  [NEXT/ALL/FIRST/LAST/PREV]
                            [CHARS/PREFIX/SUFFIX/WORD]
                            [X/NX]
                            [col-1 [col-2]]
```

FIND/CHANGE STRINGS (BROWSE AND EDIT)

Character String Format:

Simple string:	cccccc	(no embedded blanks or commas)
Delimited string:	'ccccc'	or "ccccc"
Hex string:	X'hhhh'	or 'hhhh'X
Text string:	T'cccc'	or 'cccc'T
Picture string:	P'ssss'	or 'ssss'P
Previous string:	*	(single asterisk)

Picture Strings - Special Characters:

P'='	- any character (don't care)
P'¬'	- any non-blank character
P'.'	- any non-displayable (invalid) character
P'#'	- any numeric character (0-9)
P'-'	- any non-numeric character
P'@'	- any alphabetic character (upper or lower case)
P'<'	- any lower case alphabetic character
P'>'	- any upper case alphabetic character
P'$'	- any special character (not alpha or numeric)

APPENDIX F: SPF/TSO - SPF/CMS DIFFERENCES

SPF/CMS is functionally equivalent to the SPF/TSO program product, 5740-XT8, Version 2.2. The two products are fully compatible in display formats and operation, except for those features which are explicitly oriented to the VS2/TSO or VM/CMS environment. The following is a list of those differences.

- On the primary option menu, option 6 is:

  SPF/TSO: Enter TSO Command or CLIST
  SPF/CMS: Enter CMS Command or EXEC

- Menus and tutorials which refer to "data sets" in SPF/TSO refer to "files" in SPF/CMS. The first part of the data set (file) menus allows specification of an SPF library. The format is identical in SPF/TSO and SPF/CMS, namely:

```
SPF LIBRARY:
   PROJECT ===>
   LIBRARY ===>
   TYPE    ===>
   MEMBER  ===>          (BLANK FOR MEMBER SELECTION LIST)
```

  The second part of the data set (file) menus differs as follows. For SPF/TSO:

```
OTHER PARTITIONED OR SEQUENTIAL DATASET:
   DATASET NAME  ===>
   VOLUME SERIAL ===>          (IF NOT CATALOGED)
```

  For SPF/CMS:

```
CMS FILE:
   FILE ID ===>
   MEMBER  ===>          (FOR MACLIB OR TXTLIB)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>       DEVICE ADDR. ===>     LINK ACCESS MODE ===>
```

  The third part of the data set (file) menus contains an (OS) password field for SPF/TSO. For SPF/CMS, this part of the menu may contain a read password field, an update password field, or both, depending on the type of function. In SPF/TSO, passwords pertain to data sets; in SPF/CMS, they pertain to minidisks.

- Implementation of SPF libraries in SPF/TSO is accomplished using partitioned data sets, while SPF/CMS uses sequential CMS files and/or MACLIBs and TXTLIBs (under user option).

- SPF/TSO supports printing on either:

  System printer (via submission of a background job)
  328x printer (via interface to DSPRINT command)

  SPF/CMS supports printing via the CMS PRINT command. The output may be directed to the system printer or spooled to another user (or machine), which may be at a remote node in the network. SPF/CMS does not support printing on a 328x printer.

- Specification of "job statement information" occurs in SPF/TSO on the background selection menu and all menus pertaining to hardcopy output. SPF/CMS provides job information on the batch selection menu only. All other output is accomplished with the CMS PRINT, PUNCH, or DISK DUMP commands, rather than via job submission.

Additionally, job statement information syntax and keywords are different, reflecting differences between CMS batch job streams and VS2 JCL.

- Differences in foreground processing (option 4) and background (batch) processing (option 5) are as follows:

  SPF/TSO: Compiled/assembled output (object module) goes into an SPF library whose "type" is OBJ. Link edit output (load module) goes into an SPF library whose "type" is LOAD.

  SPF/CMS: Compiled/assembled output (object module) goes into an SPF library whose "type" is TEXT. Linkage editing is not supported, but the LOAD command is supported.

- Option 3.2 (dataset/file utility). The following suboptions of this utility in SPF/TSO:

  A - Allocate new data set
  C - Catalog data set
  U - Uncatalog data set

  are replaced in SPF/CMS with:

  S - Specify new SPF library
  U - Unspecify SPF library

  There is no need for an "allocate file" capability under SPF/CMS, since space allocation is handled automatically by VM/CMS. There is, however, a need to pre-specify SPF library identifiers and file characteristics to SPF/CMS.

- Option 3.4. In SPF/TSO this is the catalog management utility. In SPF/CMS it is replaced with the project utility, which provides a similar function for SPF libraries.

- Option 3.6. In SPF/TSO this is the hardcopy utility. In SPF/CMS it is replaced with the spool utility, which provides similar functions for printing and punching, and also allows spooling to another user (or machine), which may be at a remote node in the network.

- Option 3.7. In SPF/TSO this is the list VTOC utility, which has no equivalent in SPF/CMS. It is replaced with a new SPF/CMS function, the reader utility.

- Option 3.8. In SPF/TSO this is the outlist utility, which has no equivalent in SPF/CMS. It is replaced with a new SPF/CMS function, the retrieve utility.

INDEX

Structured Programming Facility/Conversational Monitor System: Program Reference Manual Printed in U.S.A. SH20-2409-0

SH20-2409-0

2/072/980

Structured Programming Facility/Conversational Monitor System:
Program Reference Manual
SH20-2409-0

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ______________________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SH20-2409-0

Reader's Comment Form

Fold and tape — **Please Do Not Staple** — Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 824
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape — **Please Do Not Staple** — Fold and tape

IBM®

Cut or Fold Along Line

Structured Programming Facility/Conversational Monitor System: Program Reference Manual Printed in U.S.A. SH20-2409-0