Order Number SC34-2090-0
File Number S370/4300-39

Program Product

Interactive System Productivity Facility/ Program Development Facility for VM/SP: Program Reference

IBM

Order Number SC34-2090-0
File Number S370/4300-39

Program Product

Interactive System Productivity Facility/ Program Development Facility for VM/SP: Program Reference

Program Number 5664-172

First Edition (May, 1982)

This edition applies to the Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) Program Product, Program Number 5664-172, for use with VM/SP, and to all subsequent releases until otherwise indicated by Technical Newsletters.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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 reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Systems Publications, Dept Z59, Bldg 931, P.O. Box 390, Poughkeepsie, New York U.S.A 12602. 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.

PREFACE

The Interactive System Productivity Facility (ISPF) and the ISPF/Program Development Facility (ISPF/PDF or PDF) are related IBM program products. Together, they are designed to improve user productivity in the development of applications, and contain special functions for the development, test, and use of interactive applications, called dialogs. Specifically:

- ISPF is a dialog manager for interactive applications. It provides control and services to support execution of dialogs.

- PDF is a facility that aids in the development of dialogs and other types of applications. It makes use of display terminals and an interactive environment to assist with many programming tasks.

This manual tells you how to use the PDF under VM/SP. It assumes that you are an application or systems programmer, engaged in program development, and are familiar with the VM environment.

The first three chapters of this manual provide you with information pertinent to the general use of PDF -- use of the terminal, format of the displays, a description of the programming libraries, etc. Each succeeding chapter deals with the use of a particular PDF function, as selectable from the primary option menu. Finally, the appendixes contain a usage scenario, a description of listing formats, and command summaries.

The dialog management functions of ISPF and the program development functions of PDF were previously combined in the predecessor program product, the System Productivity Facility (SPF). The major changes from SPF are described in ISPF General Information, GC34-2078. You may also want to refer to the following publications:

ISPF Dialog Management Services (SC34-2088) - Provides a detailed description of the dialog management services and related information required to develop an interactive application that runs under ISPF.

ISPF Dialog Management Services Examples (SC34-2085) - Provides a set of examples of the use of dialog management services.

ISPF for VM/SP Installation and Customization (SC34-2083) - Provides information needed to install ISPF and ISPF/PDF, and to custom tailor these products for a particular installation.

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

- Uppercase commands and their operands should be entered as shown, but not necessarily in uppercase. Operands shown in lowercase are variable; you substitute your own value for them.

- Operands shown in brackets [ ] are optional, with a choice indicated by stacking the operands. You may choose one or none; the defaults are underscored.

- Operands shown without brackets are required. If several required operands are stacked, you must select one of the choices from the stack.

CONTENTS

FIGURES

CHAPTER 1. INTRODUCTION

The ISPF/Program Development Facility (ISPF/PDF, or PDF) Program Product is an interactive dialog application that you can use to simplify many of your programming tasks. PDF may be used either by individual programmers, each working on a separate project, or by a group of programmers working on a common project.

MAJOR FUNCTIONS

PDF is an ISPF dialog that you use at a display terminal. The functions available to you include:

- **Multilevel programming library support** that makes it easier to maintain and track program segments at different versions or levels.
- **Full-screen, context editing** that allows multiple changes to information on a screen with one interaction with the host system. The most frequently used editing functions are invoked using simple, one-character commands.
- **Models** that assist in the development of dialog panels, messages, function routines, file skeletons, and tables.
- **Scrolling** of source data and listings in any direction. Also, location of data by character string or line number.
- **Split screen** capability that allows you to partition the display screen into two "logical" areas.
- **Utilities** to specify and maintain libraries and files.
- **Interface to standard language processors** (compilers, assemblers, and a loader). These processors may be invoked either in the foreground or as CMS batch jobs.
- **Dialog test** facilities that assist in the development and testing of ISPF dialog applications.
- **Documentation preparation assistance** consisting of text editing facilities, and an interface to the Document Composition Facility Program Product.
- **Online tutorial** for instruction and reference. This feature is especially valuable if you are a new or an occasional user.

STRUCTURED PROGRAMMING

PDF permits use of both structured and conventional programming techniques.

While conventional programming consists of large blocks of unindented code, structured programming emphasizes the use of segmentation and indentation. A structured source program normally consists of a large number of relatively small segments, with each segment stored as a separate member 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. Functions that are specifically oriented toward structured programming include:

- Segmentation. One segment (member) can easily be split into multiple segments, or multiple segments can be merged into one.

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

- Insertion 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. A block of code may be temporarily excluded from display so that the space that it normally occupies on the screen is closed up. This makes it easy for you to see the control structure, particularly when the length of a segment exceeds the screen size.

INTERACTIVE APPLICATIONS

PDF aids in the development of interactive applications, called **dialogs**, that run under the ISPF dialog manager. Models are provided to help you create the various ISPF elements of a dialog application, and the following functions are available to help you test the dialogs:

- Dialog functions may be invoked, panels displayed, variable data displayed and manipulated, table data displayed and changed, and ISPF services called. These facilities are available before you begin executing a dialog or at a breakpoint during testing.

- You may stop execution of the dialog at any call to an ISPF service by establishing a breakpoint. At a breakpoint, dialog data (variable data, table data, etc.) may be displayed and changed.

- You may trace the flow of the dialog's calls to ISPF services and the usage of variables. Trace information is placed in the ISPF log.

USAGE SCENARIO

The usage scenario shown in Appendix A is a brief scenario of PDF terminal operations. It is intended to serve as a demonstration that PDF has been properly installed and is operational. It may also be useful for new users as a quick introduction to PDF.

CHAPTER 2. INTERACTING WITH PDF

This chapter tells you how to begin and end your PDF session, and how to use the display screen and terminal keys.

PANEL DISPLAYS

PDF communicates with you through a series of panels. A **panel** is a predefined display image that is presented on a display terminal. Panels may require a response, and that response is used to determine the next panel to be displayed or the function to be performed. Your responses may be in uppercase or lowercase.

Format of Panels

All panels are formatted to fit on a 24-line by 80-character screen. On a 3278 Model 3 or 4, scrollable data occupies the full length of the screen (32 or 43 lines).

On a 3278 Model 5, information is normally displayed in "default" mode (24 lines by 80 characters) with the same size characters as other models. Browse and edit data that is wider than 80 characters is displayed with the smaller "native" mode characters (up to 132 per line). You may override the automatic switching of modes using option 0.1.

The first three lines of each display are formatted as follows:

line 1	Title	Short Message
line 2	Command/Option	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 (browse) and column positions (browse and edit)

- Successful completion of a processing function

- Error conditions (accompanied by audible alarm, if installed)

The command/option area (line 2) is where you enter a command or an option selection. The scroll area (line 2) contains the current scroll amount whenever scrolling is applicable. You may overtype the scroll amount.

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

Types of Panels

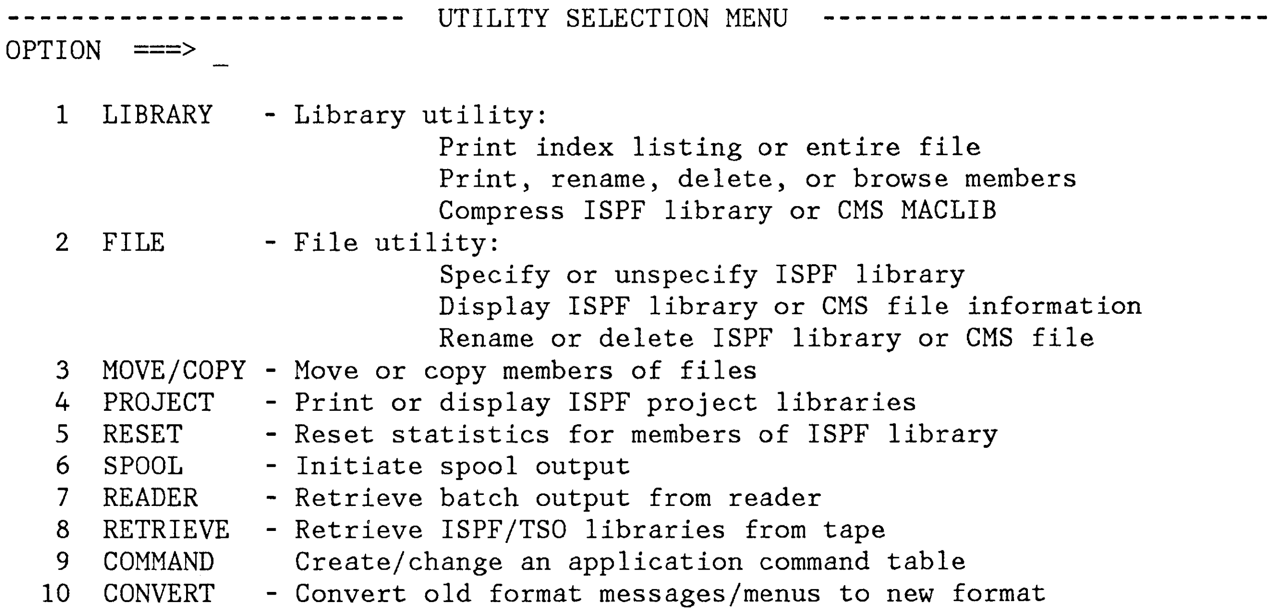
In using PDF, you will see four basic types of panels:

1. Selection Panel - You select from a list of options by typing a code and pressing the ENTER key. Selection panels are also called menus. Figure 1 shows an example: the panel for selecting a PDF utility.

2. Entry Panel - You supply parameters by filling in labeled fields. In many cases, fields are pre-entered based on what you last entered. Figure 2 shows an example: the panel on which you identify the data to be browsed.

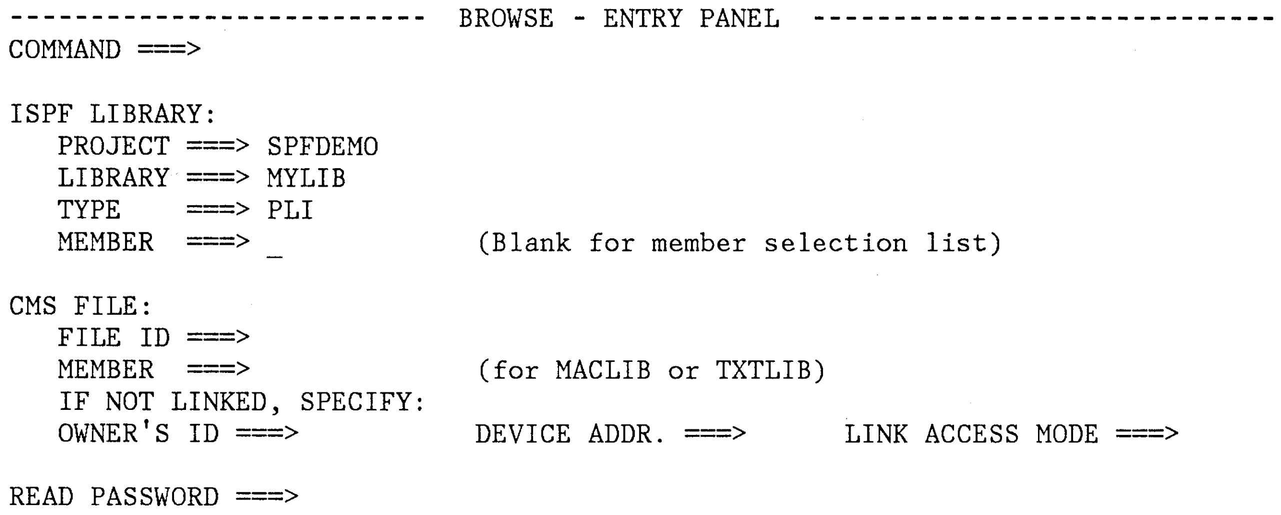
3. Member List - Displays a list of members in a programming library. You may select a member by entering a one-character code in front of the appropriate member name. Figure 3 shows an example: a member list on which you have selected the member named COINS.

4. Scrollable Data Display - Displays source code or output listings. Figure 4 shows an example: the browse display of source code.



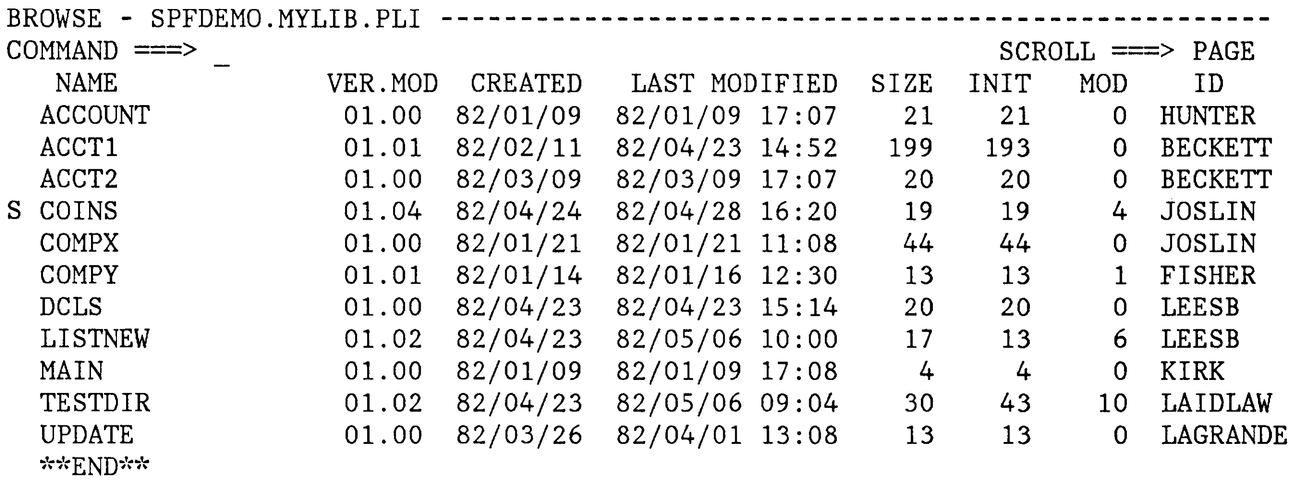
```
------------------------  UTILITY SELECTION MENU  ---------------------------
OPTION  ===> _

   1  LIBRARY   - Library utility:
                           Print index listing or entire file
                           Print, rename, delete, or browse members
                           Compress ISPF library or CMS MACLIB
   2  FILE      - File utility:
                           Specify or unspecify ISPF library
                           Display ISPF library or CMS file information
                           Rename or delete ISPF library or CMS file
   3  MOVE/COPY - Move or copy members of files
   4  PROJECT   - Print or display ISPF project libraries
   5  RESET     - Reset statistics for members of ISPF library
   6  SPOOL     - Initiate spool output
   7  READER    - Retrieve batch output from reader
   8  RETRIEVE  - Retrieve ISPF/TSO libraries from tape
   9  COMMAND     Create/change an application command table
  10  CONVERT   - Convert old format messages/menus to new format
```

Figure 1. Utility Selection Panel



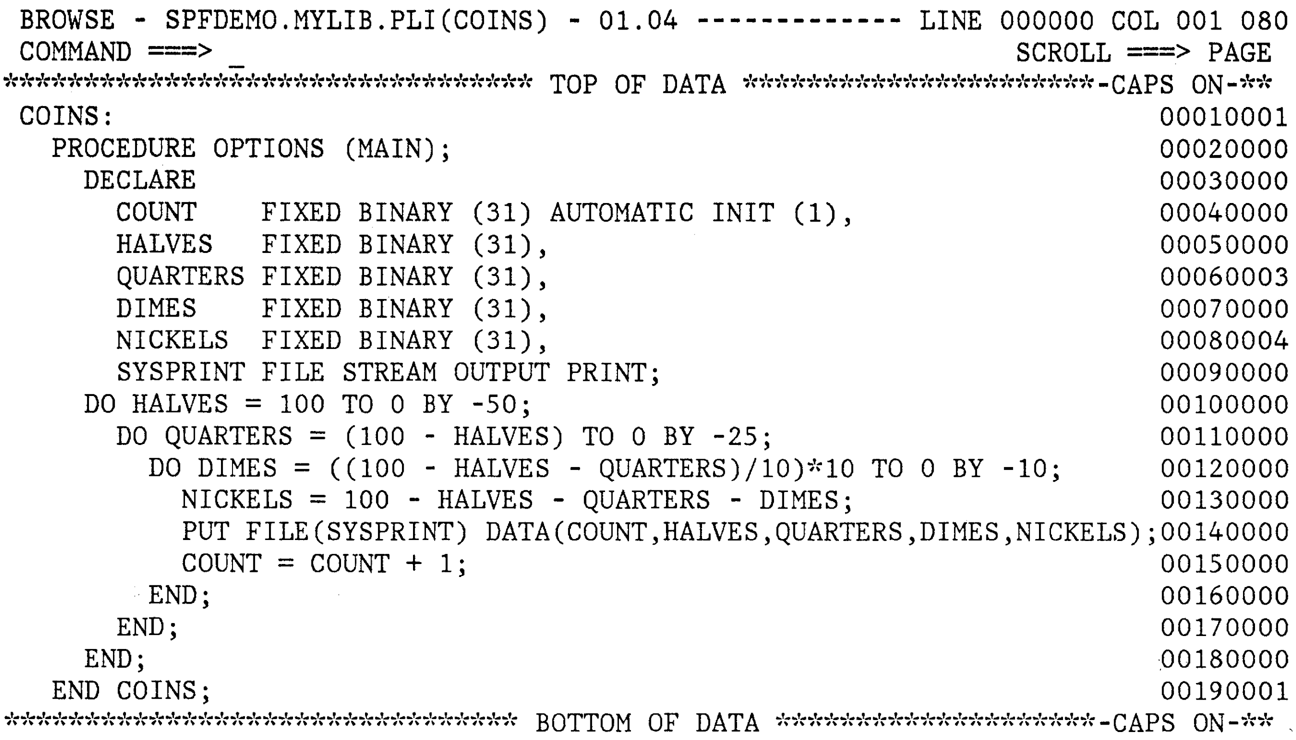
```
-------------------------  BROWSE - ENTRY PANEL  ----------------------------
COMMAND ===>

ISPF 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 - Entry Panel



```
BROWSE - SPFDEMO.MYLIB.PLI ------------------------------------------------------
COMMAND ===> _                                                 SCROLL ===> PAGE
   NAME             VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT            01.00  82/01/09  82/01/09 17:07    21    21     0  HUNTER
  ACCT1              01.01  82/02/11  82/04/23 14:52   199   193     0  BECKETT
  ACCT2              01.00  82/03/09  82/03/09 17:07    20    20     0  BECKETT
S COINS              01.04  82/04/24  82/04/28 16:20    19    19     4  JOSLIN
  COMPX              01.00  82/01/21  82/01/21 11:08    44    44     0  JOSLIN
  COMPY              01.01  82/01/14  82/01/16 12:30    13    13     1  FISHER
  DCLS               01.00  82/04/23  82/04/23 15:14    20    20     0  LEESB
  LISTNEW            01.02  82/04/23  82/05/06 10:00    17    13     6  LEESB
  MAIN               01.00  82/01/09  82/01/09 17:08     4     4     0  KIRK
  TESTDIR            01.02  82/04/23  82/05/06 09:04    30    43    10  LAIDLAW
  UPDATE             01.00  82/03/26  82/04/01 13:08    13    13     0  LAGRANDE
  **END**
```

Figure 3. Browse - Member List



```
 BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------- LINE 000000 COL 001 080
 COMMAND ===> _                                                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-**
```

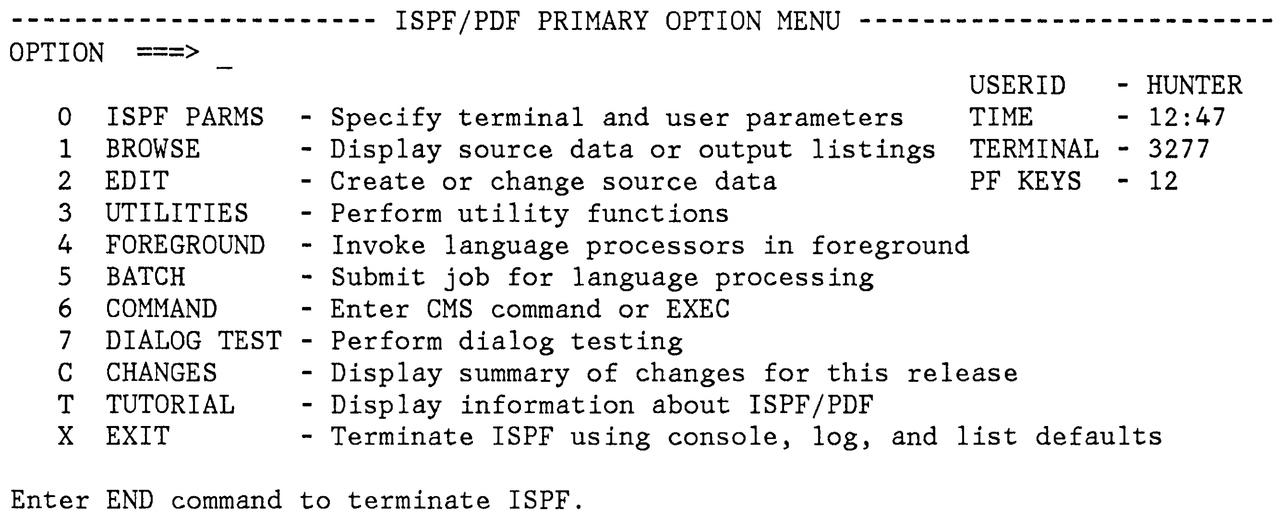
Figure 4. Browse - Data Display

PDF INVOCATION

Before PDF is invoked, you must access your virtual device 191 as the A-disk. PDF assumes that this minidisk is available at all times in read/write mode, and that no other user has write access to it. To invoke PDF, enter the command "PDF". (Your installation may establish an alias for PDF, such as ISPF or SPF.) The first panel to be displayed is the primary option menu (Figure 5). You may select an option by typing a one-character code in the option field and pressing the ENTER key; for example:

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

to select the edit option.



```
---------------------- ISPF/PDF PRIMARY OPTION MENU -------------------------
OPTION  ===> _
                                                         USERID   - HUNTER
   0  ISPF PARMS  - Specify terminal and user 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  UTILITIES   - Perform utility functions
   4  FOREGROUND  - Invoke language processors in foreground
   5  BATCH       - Submit job for language processing
   6  COMMAND     - Enter CMS command or EXEC
   7  DIALOG TEST - Perform dialog testing
   C  CHANGES     - Display summary of changes for this release
   T  TUTORIAL    - Display information about ISPF/PDF
   X  EXIT        - Terminate ISPF using console, log, and list defaults

 Enter END command to terminate ISPF.
```

Figure 5. Primary Option Menu

For options that have secondary selection panels (options 0, 3, 4, 5, and 7), you may bypass the second selection panel by typing two numbers, separated by a decimal point, on the primary option menu. For example, entering "3.1" on the primary option menu has the same effect as entering "3" on the primary option menu and "1" on the secondary selection panel.

On initial entry, you may also bypass the primary option menu (and the secondary selection panel) by entering an initial option as a parameter of the PDF command. Examples:

```
PDF OPT(2)     - to go directly to edit
PDF OPT(3.1)   - to go directly to utility suboption 1
```

The PDF primary options are:

0 ISPF PARMS To specify terminal and user parameters and defaults, including: terminal type, number of program function (PF) keys and PF key definitions, default pad character for input fields, and defaults for processing console, log, and list files.

1 BROWSE

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

2 EDIT

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

3 UTILITIES

To print, rename, or delete ISPF library members or CMS files; specify ISPF libraries; move or copy data; display or print ISPF project listings; reset ISPF library statistics; initiate spool output; retrieve data from the virtual reader; retrieve ISPF libraries (via tape) from an MVS system; create or modify application command tables; or convert menus and messages to ISPF format from the old SPF format.

4 FOREGROUND

To execute language processing programs in the foreground, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL/VS, linkage editor or CMS loader, COBOL or FORTRAN interactive debug, and the SCRIPT/VS document formatter.

5 BATCH

To generate and submit job statements and command streams for batch execution of IBM language processing programs, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL, linkage editor and CMS loader.

6 COMMAND

To enter a CMS or CP command or an EXEC during execution of PDF.

7 DIALOG TEST

To test a panel or dialog function; display a panel or message; set or examine dialog variables; set up, modify, or examine tables; browse the ISPF log; set up or modify breakpoints; and set up or modify traces of functions or variables.

C CHANGES

To obtain online instruction about the changes between the System Productivity Facility program development facility and ISPF/Program Development Facility.

T TUTORIAL

To obtain immediate online instruction in the use of PDF. The tutorial may be viewed sequentially from beginning to end, or randomly by selecting topics from its table of contents or index. The tutorial may also be entered from other PDF options using the HELP command.

X EXIT To exit from PDF using the console, log, and list defaults defined using option 0.2.

COMMAND ENTRY

PDF commands are provided for commonly used functions. You can enter a PDF command in one of two ways:

1. By keying the command in the command field and pressing ENTER. (This includes the command field in browse, edit, member lists, and table displays, as well as the command/option field on a panel.)

2. By pressing a program function (PF) key.

There are no special functions associated with PF keys. A PF key simply simulates command entry. Each PF key is equated to a character string. When you press the PF key, the processing is the same as if you typed the character string in the command field and pressed the ENTER key. ISPF does not sense the difference between a command entered with a PF key and a command entered by typing in the command field.

Before you press a PF key, you may enter information in the command field. If so, the PF key definition is concatenated ahead of the contents of the command field. For example, suppose you equate PF7 to the character string "UP". If you type "4" in the command field and then press PF7, the results are exactly the same as if you had typed "UP 4" in the command field and then pressed ENTER.

You can stack commands for execution by entering a special delimiter between the commands. The default delimiter is a semicolon (;). You can change the default using the ISPF Parms option (option 0.1). For example:

```
  COMMAND ===> CHANGE ALL ABC XYZ;CHANGE ALL PQR GHIJK
```

In this example, two edit CHANGE commands have been stacked.

The PDF commands include all the functions that were available in previous SPF products only by using PF keys. These PDF commands are:

```
HELP      SPLIT      END
RETURN    RFIND      RCHANGE
UP        DOWN       SWAP
LEFT      RIGHT      CURSOR
PRINT     PRINT-HI
```

PDF also includes the following new commands: CP or CMS, KEYS, and PANELID.

> **Note:** The command designations for the FIND/CHANGE "key" functions have been changed to RFIND (repeat find) and RCHANGE

(repeat change), to avoid confusion with the FIND and CHANGE commands that are used in browse and edit.

Following is a description of the general PDF commands. Additional commands are defined for member lists, browse, edit, and dialog test. The default PF key definitions are shown following the command name.

HELP (PF1/13) Displays additional information about an error message or provides tutorial information about PDF commands and options.

SPLIT (PF2/14) Causes split screen mode to be entered, or changes the location of the split line.

END (PF3/15) Terminates the current operation and returns to the previous panel. If the primary option menu is displayed, this command terminates PDF.

RETURN (PF4/16) Causes an immediate return to the primary option menu or to the panel from which a HELP or KEYS command was entered, without displaying the intervening panels. Note that the dialog test primary option menu (displayed when option 7 is selected) is itself a primary option menu; when you are in option 7, the RETURN command stops at that menu. For other uses of the RETURN command, see the section entitled "Jump Function."

RFIND (PF5/17) Repeats the action of the previous FIND command or the FIND part of the most recent CHANGE command (applies to browse and edit only).

RCHANGE (PF6/18) Repeats the action of the previous CHANGE command (applies to edit only).

UP (PF7/19) Causes a scroll toward the top of the data.

DOWN (PF8/20) Causes a scroll toward the bottom of the data.

SWAP (PF9/21) Moves the cursor to wherever it was previously positioned on the other logical screen of a split screen pair.

LEFT (PF10/22) Causes a scroll toward the left margin of the data.

RIGHT (PF11/23) Causes a scroll toward the right margin of the data.

CURSOR (PF12/24) Moves the cursor to the first input field on line 2 (normally the option selection or command field). Entering this command again causes the cursor to be moved to the second input field on line 2, if any (normally the scroll amount field).

PRINT

Causes a "snapshot" of the screen image to be recorded in the ISPF list file, for subsequent printing.

PRINT-HI

Same as PRINT except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. Do not use this command if the list output is to be spooled to a printer that does not have a suppress-space carriage control.

CP or CMS

Allows you to enter a CP or CMS command or EXEC from any panel. For example:

```
COMMAND ===> CMS LISTFILE * * B
```

Note: Under ISPF, the CP and CMS commands are synonymous. Either one may be used to invoke a CP command, a CMS command, or an EXEC.

KEYS

Causes an immediate display of a panel that allows you to view and change the current PF key definitions (equivalent to option 0.3).

PANELID

The command **PANELID** or **PANELID ON** causes all subsequent panels to be displayed with the identifier of the panel shown at the beginning of line 1 (provided the panel contains a protected-field attribute byte in row 1 column 1, and no other attribute bytes in the next eight character positions). The command **PANELID OFF** suppresses display of the panel identifiers. The OFF mode is in effect at the beginning of a PDF session.

Note: The SPLIT, RFIND, RCHANGE, SWAP, and CURSOR commands are sensitive to the current cursor position and are most easily entered using PF keys. The four scrolling commands are also cursor-sensitive if the scroll amount is set to "CSR". See "Scrolling."

JUMP FUNCTION

The jump function allows you to move from one option to another without displaying the primary option menu, as follows: In the command field on any panel, or in any input field preceded by an arrow, enter an equal sign (=) immediately followed by a primary option. Then press the ENTER key or the RETURN PF key.

For example, suppose you are in edit and you wish to use the library utility (option 3.1). Enter:

```
COMMAND ===> =3.1
```

This causes edit to be ended and the library utility panel to be displayed. The processing is exactly the same as if you had repeatedly entered END commands until the primary option menu was displayed, and then you had entered 3.1 on the primary option menu.

Note the following points:

- In the previous SPF product, this facility was called "extended return" and required the use of the RETURN PF key. For compatibility, you may still use the RETURN PF key when you enter the jump function, but use of the ENTER key is recommended.

- You may enter "=X", which is equivalent to selecting the X (exit) option on the primary option menu. Except under option 7, "=X" immediately terminates:

  - ISPF/PDF if you are not in split screen mode

  - The logical screen if you are in split screen mode

- Under option 7 (dialog test), the jump function pertains to options that may be selected from the dialog test primary option menu. For example, suppose you are in option 7.3. If you enter "=1", you go to option 7.1. On the other hand, if you enter "=X", you exit dialog test and return to the ISPF/PDF primary option menu.

- Unlike the RETURN command, the jump function does not stop upon completion of the HELP or KEYS command. For example, suppose you are in edit and you enter a HELP command, which takes you into the tutorial. From the tutorial, if you enter RETURN (without the jump function), you return to the edit display from which you entered HELP. But if you enter "=3.1", you leave the tutorial, end edit, and go directly to the library utility.

TERMINAL KEYS

On the terminal, the program access (PA) and program function (PF) keys are used to request commonly used operations. No PF keys are required for PDF operations, but PDF has a default set of PF key definitions that you can change. Keys that are not used for general PDF operations may be equated to edit, browse, and dialog test primary commands, to edit and dialog test line commands, or to CMS commands, using the parms option (option 0.3) or the KEYS command.

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

Program Access Keys

The two PA keys are defined as follows. These definitions may not be changed.

ATTENTION (PA1) Causes an immediate return to CP mode. You should not use this key, since it bypasses normal PDF termination. If you press it by mistake, enter a BEGIN command to resume PDF.

RESHOW (PA2) Redisplays the contents of the screen. PA2 may be useful if you pressed the ERASE INPUT or CLEAR key accidentally, or if you typed unwanted information but have not yet pressed ENTER or a PF key. Note that PDF does not allow use of the Field Mark character (same key as PA2).

Program Function Keys

The PDF default PF key assignments for the 3-key by 4-key pad (right-hand side of the keyboard) are shown in Figure 6. These are PF keys 1-12 on a 12-key terminal, or keys 13-24 on a 24-key terminal.

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 general commands, and redefine PF keys 1-12 to specific operations for edit, browse, or dialog test.

PA1 ATTENTION	PA2 RESHOW

PF1 / 13 HELP	PF2 / 14 SPLIT	PF3 / 15 END
PF4 / 16 RETURN	PF5 / 17 RFIND	PF6 / 18 RCHANGE
PF7 / 19 ↑ (UP)	PF8 / 20 ↓ (DOWN)	PF9 / 21 SWAP
PF10 / 22 < (LEFT)	PF11 / 23 > (RIGHT)	PF12 / 24 CURSOR

Figure 6. Default Program Key Arrangement

SCROLLING

During edit and browse, you are generally dealing with information that exceeds the screen size. Scrolling allows you to move the screen "window" up, down, left, or right across the information. You can also scroll a member list up and down if it exceeds a single screen length.

You use four commands for scrolling (UP, DOWN, LEFT, and RIGHT). Whenever scrolling is allowed, a scroll amount is displayed at the top of the screen (line 2). This value indicates the number of lines (or columns) scrolled with each use of a scroll command. To change the scroll amount, move the cursor to the scroll field and overtype 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 scrolling command 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 scrolling command 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 occurs.

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

For scrolling purposes, a "page" is defined as the amount of information currently visible on the logical screen. In split screen mode, for example, a browse display might have 12 80-column lines of scrollable data. In this case, a scroll amount of HALF would move the window up or down by 6 lines, or right or left by 40 columns.

The current scroll amount is saved in the user profile. Three different values are saved -- one for browse, one for edit, and one for member lists. When you overtype the scroll amount, the new value remains in effect for that type of operation until you change it again. The value MAX is an exception; following a MAX scroll, the scroll amount reverts to its previous value.

You can also enter any valid scroll amount in the command field and use it with a scrolling command or PF key. For example, you can enter:

```
COMMAND ===> UP 3
```

and press the ENTER key or

```
COMMAND ===> 3
```

and press the UP PF key. Either form results in a temporary, one-time override of the scroll amount.

OVERTYPING OF SCROLL AND MULTIPLE CHOICE PARAMETERS

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

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

- To change the scroll amount to a number of lines or columns, overtype the first character(s) with the desired number. Any alphabetic characters following a number are ignored. For example, "3AGE" is interpreted as "3".

The same rule applies to multiple choice 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, overtype the first character with "Y" or "M", respectively.

SPLIT SCREEN

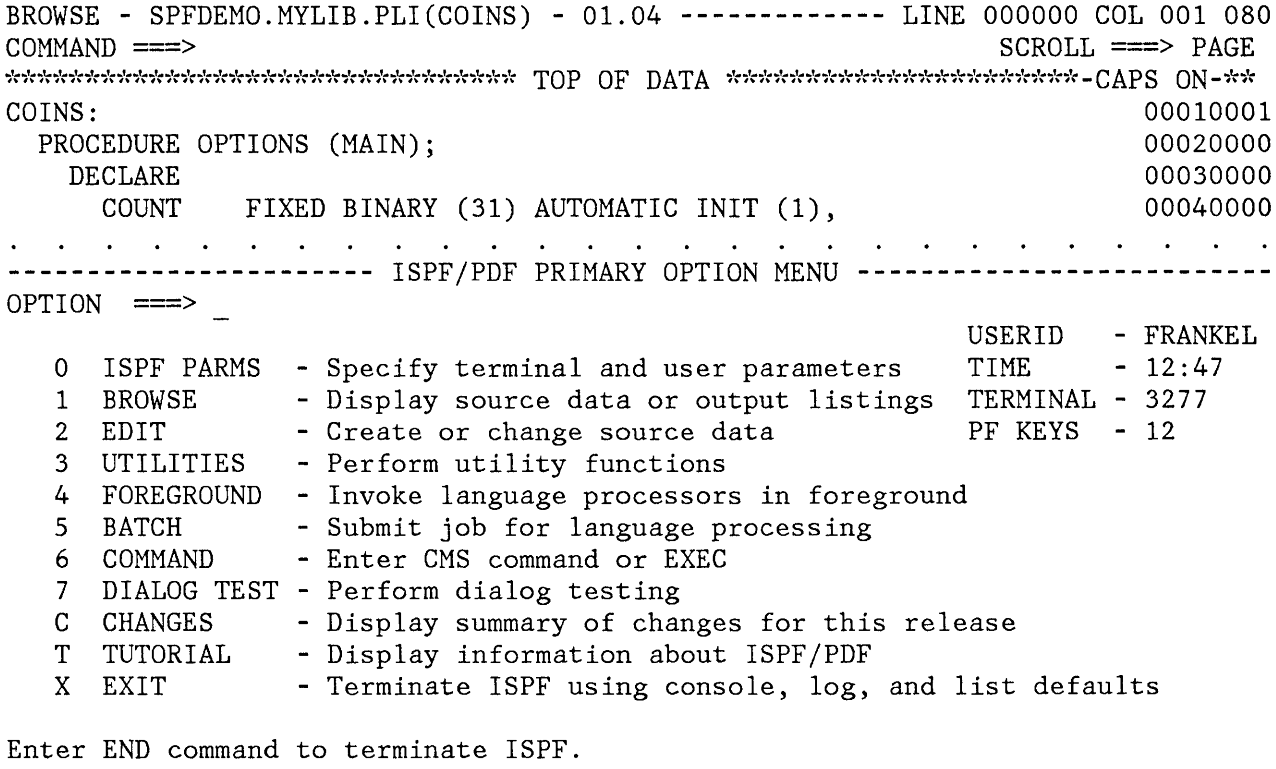
Split screen mode allows you to partition the display screen into two logical screens. The top and bottom screens are treated as though they were displayed on independent terminals. Functions that can be performed simultaneously on two separate terminals can be done on the top and bottom screens. Functions that are in conflict on two separate terminals (editing the same CMS sequential file, or browsing the same CMS file) will also be in conflict when attempted simultaneously on the top and bottom screens. Since only one logical screen is active at a time, you cannot split the screen while executing a foreground compilation or a CMS command.

You can enter split screen mode by moving the cursor to the desired line for the split and then pressing the SPLIT PF key. Alternatively, you can type SPLIT in the command/option field and then move the cursor to the desired line before pressing the ENTER key. In either case, the line containing the cursor becomes the split line and is identified by a row of periods. All lines below the cursor are treated as the bottom screen. The first display to appear on the bottom screen is always the primary option menu. An example is shown in Figure 7.

You can reposition the split location at any time by moving the cursor to the desired line and using the SPLIT PF key (or command) again. Repositioning may be required to bring an entire panel or tutorial page into view.

In split screen mode, either one or the other of the logical screens is considered active at any one 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 identifies which of the two screens is active. To switch from one screen to the other, use the cursor movement keys or enter the SWAP command.

In addition to moving the cursor to the other logical screen, the SWAP command automatically repositions 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, SWAP moves the cursor to the top screen and repositions the split to the third line up from the bottom. This allows the use of two screens in "flip-flop" mode, with each logical screen consuming nearly all lines of the physical screen.



```
BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------- LINE 000000 COL 001 080
COMMAND ===>                                                  SCROLL ===> PAGE
******************************** TOP OF DATA ***********************-CAPS ON-**
COINS:                                                                 00010001
  PROCEDURE OPTIONS (MAIN);                                            00020000
    DECLARE                                                            00030000
      COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),                   00040000
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
---------------------- ISPF/PDF PRIMARY OPTION MENU ---------------------------
OPTION  ===> _
                                                            USERID   - FRANKEL
   0  ISPF PARMS  - Specify terminal and user 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  UTILITIES   - Perform utility functions
   4  FOREGROUND  - Invoke language processors in foreground
   5  BATCH       - Submit job for language processing
   6  COMMAND     - Enter CMS command or EXEC
   7  DIALOG TEST - Perform dialog testing
   C  CHANGES     - Display summary of changes for this release
   T  TUTORIAL    - Display information about ISPF/PDF
   X  EXIT        - Terminate ISPF using console, log, and list defaults

Enter END command to terminate ISPF.
```

Figure 7. Split Screen Example

To terminate split screen mode, end PDF processing on either logical screen by entering an END or RETURN command from the primary option menu, or by using "=X". The remaining logical screen is then expanded to its full size.

HELP INFORMATION

The help function allows you to obtain additional information about a message that has been displayed in the upper right corner of the screen, or general information about a PDF command or option.

If a message is displayed, enter the HELP command to cause a one-line explanation to be displayed. If this explanation is not sufficient, you can obtain further information by entering the HELP command again. This causes an entry into the appropriate section of the tutorial. If a message is not displayed, the HELP command causes a direct entry into the appropriate section of the tutorial.

Once in the tutorial, enter the END or RETURN command to return to the screen that was being viewed when the HELP command was originally entered.

PDF TERMINATION

To terminate PDF, you must have only a single logical screen displayed (see "Split Screen").

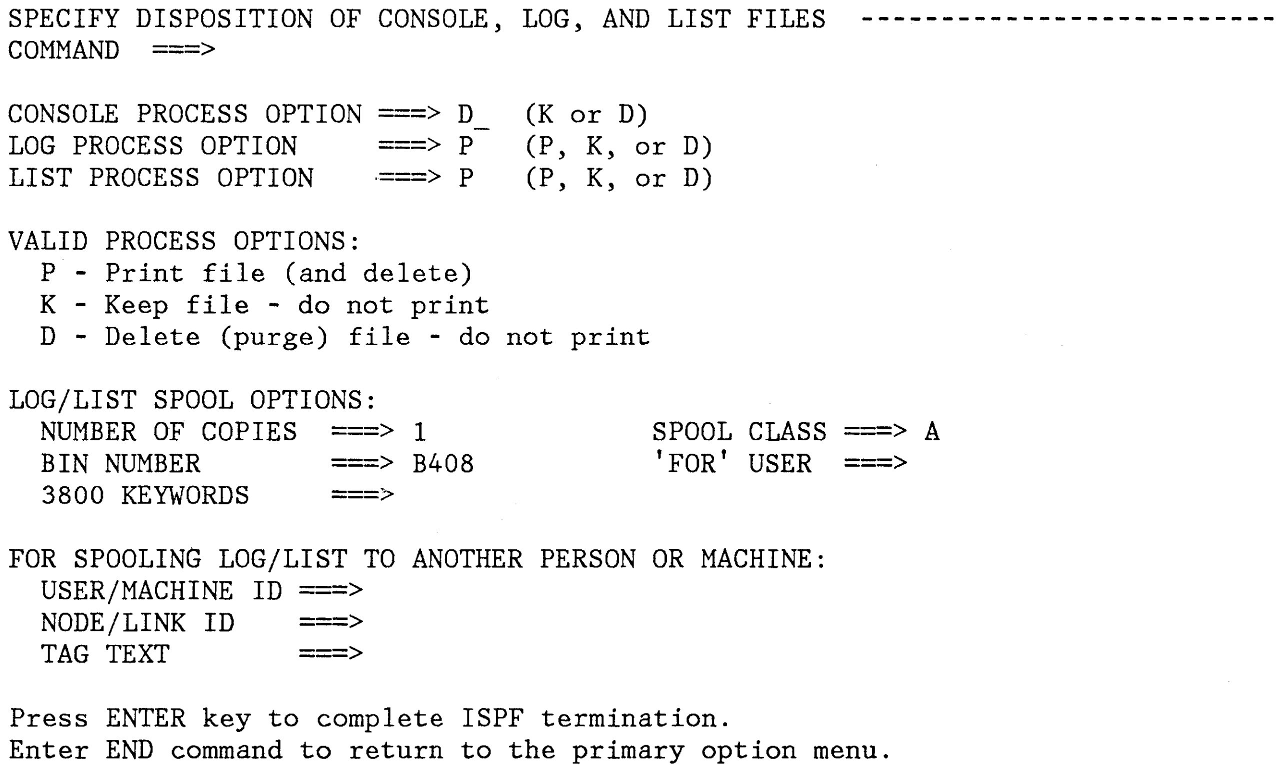
Three termination options are available from the primary option menu:

- Enter the END command to display a termination panel (Figure 8) for specification of console, log, and list processing.

- Enter option X to terminate PDF with user defaults for processing console, log, and list files, as specified using the ISPF parms option (option 0.2). If you have specified no defaults, option X causes the termination panel to be displayed.

- Enter the RETURN command. From the primary option menu, RETURN has the same effect as entering option X.

If you are not in split screen mode, you may immediately exit ISPF/PDF from any panel (except under option 7) by entering "=X". See the section entitled "Jump Function."



```
SPECIFY DISPOSITION OF CONSOLE, LOG, AND LIST FILES -------------------------
COMMAND ===>

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
  BIN NUMBER        ===> B408          'FOR' USER  ===>
  3800 KEYWORDS     ===>

FOR SPOOLING LOG/LIST TO ANOTHER PERSON OR MACHINE:
  USER/MACHINE ID ===>
  NODE/LINK ID    ===>
  TAG TEXT        ===>

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.
```

Figure 8. Termination Panel

The termination panel allows you to specify whether the virtual console is to be deleted (purged and set to "stop" status) or kept (left in "start" status), and whether the log and list files are to be printed, deleted (erased), or kept without printing.

If the log or list files (or both) are to be printed, you may specify the following spool parameters:

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

- Spool class - for other than class A.

- Bin number - for other than your default destination.

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

- 3800 keywords - if output is directed to an IBM 3800 printer that is attached to the CMS system as a virtual spooling device. See the description of the SPOOL command in VM/SP CP Command Reference for General Users, SC19-6211.

- User/machine id - to spool the output to the virtual reader of another user or machine.

- Node/link id - if the destination is a remote node on the network (valid with either "user/machine id" or "tag text", but not both).

- Tag text - to specify control parameters for printing on a remote non-VM system. See the description of the TAG command in VM/370 RSCS Networking Program Reference and Operations Manual, SH24-5005.

Once you have entered the options on the termination panel, press ENTER to complete termination. The screen is cleared, one or more termination messages are displayed, and you are returned to CMS.

CHAPTER 3. LIBRARIES AND FILES

This chapter provides the information you need to handle programming libraries, files, output listings, etc., under PDF.

ISPF LIBRARIES

An **ISPF library** is a collection of code or data units, called members. Each library generally contains members with the same type of information. Figure 9 shows a sample set of five libraries that contain Assembler source (ASSEMBLE), COBOL source (COBOL), TEXT (object) modules, test data (DATA), and SCRIPT documentation (SCRIPT).

ISPF 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.

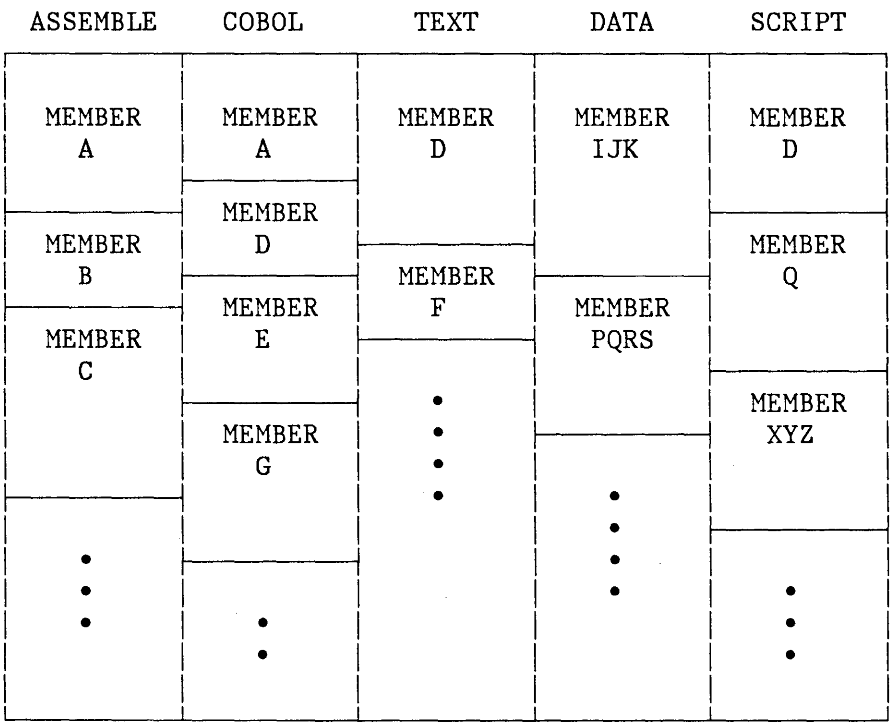


Figure 9. ISPF Library Organization

- "library name" identifies the particular set of libraries, such as MASTER, TEST, or TOM.

- "type" identifies the type of information in the library, such as ASSEMBLE, COBOL, TEXT (object modules), DATA, or SCRIPT. Except for TEXT, the type qualifier need not conform to the standard CMS naming conventions.

Most projects use a hierarchy of related libraries to maintain effective version control over the programming development process and to reduce contention in library usage. Figure 10 shows a sample 3-level hierarchy consisting of a set of master libraries, a set of test libraries, and three sets of development (private) libraries identified by user id.

New library members or members undergoing changes generally reside in the development libraries. A test library may be used to accumulate members that 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 program. Typically, a higher degree of control is maintained over the higher level libraries than over the development libraries.

PDF allows concatenation of up to four 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 10, a typical concatenation sequence might be library JOE, followed by TEST, followed by MASTER. The concatenation applies to libraries of the same type, and is restricted to libraries that belong to the same project.

The purpose of concatenation during editing is to provide for copying members to your development library. The concatenation sequence is used to search the libraries for the member to be edited. The edited member is saved in your 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 to a higher level library by means of the move/copy utility.

The purpose of concatenation 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 a compilation or assembly (object module) is stored in the lowest level TEXT library (the first library in the concatenation sequence).

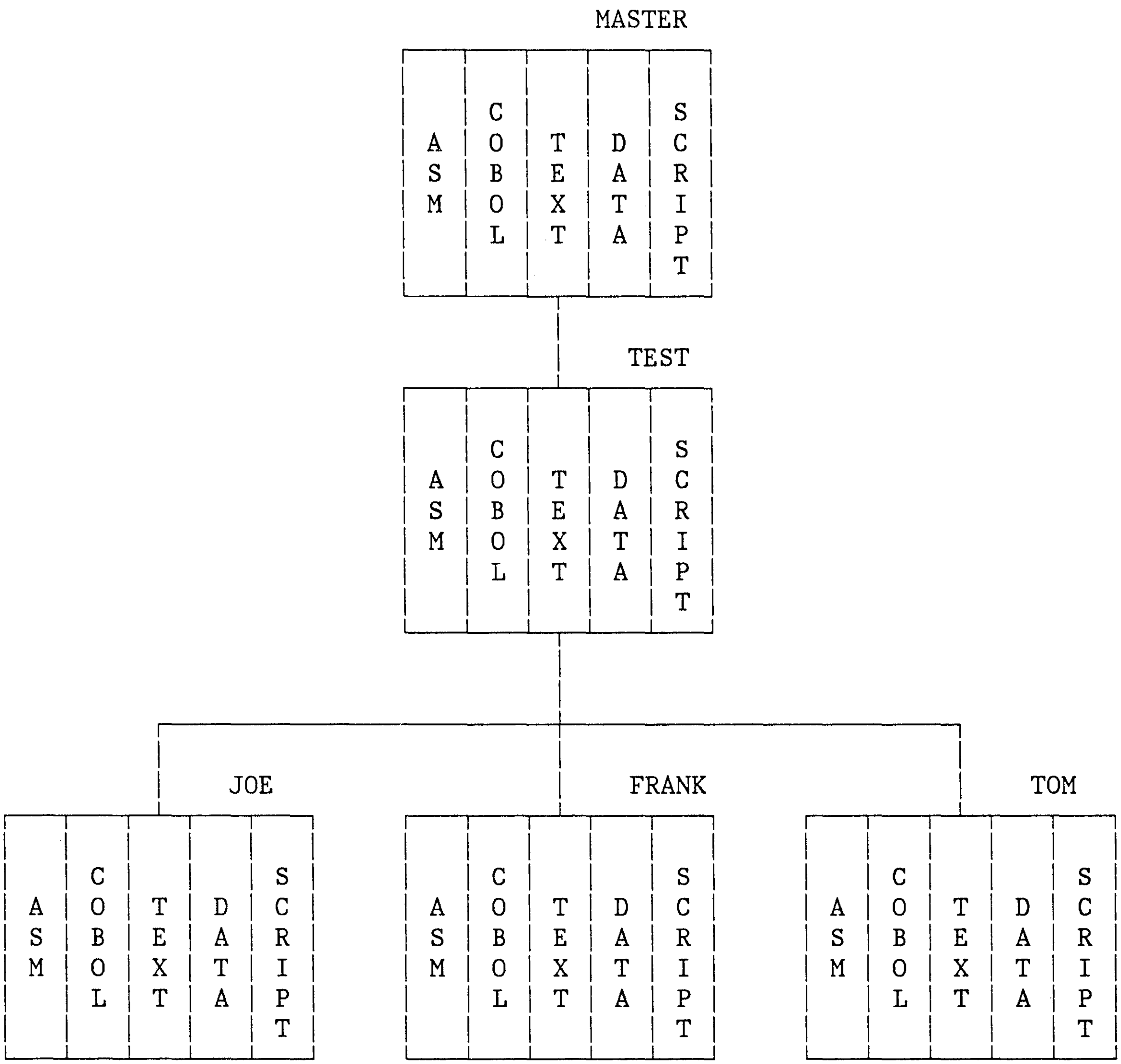


Figure 10. Hierarchy of ISPF Libraries

ISPF libraries are maintained internally as CMS files. Each 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 PDF via the file utility (option 3.2). See Appendix B for additional information about ISPF library specification and usage.

LIBRARY STATISTICS

At your option, the PDF editor automatically generates and maintains the following statistics for each member of an ISPF 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

You can change the version number and/or user id with the reset statistics utility (option 3.5). The statistics are displayed next to each member name on member lists, and may be printed by requesting an index listing via the library utility (option 3.1).

The statistics are displayed and printed with the following formats:

- 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., "82/06/27" means June 27, 1982).

- For time last modified: HH:MM (e.g., "17:20" means 5:20 PM).

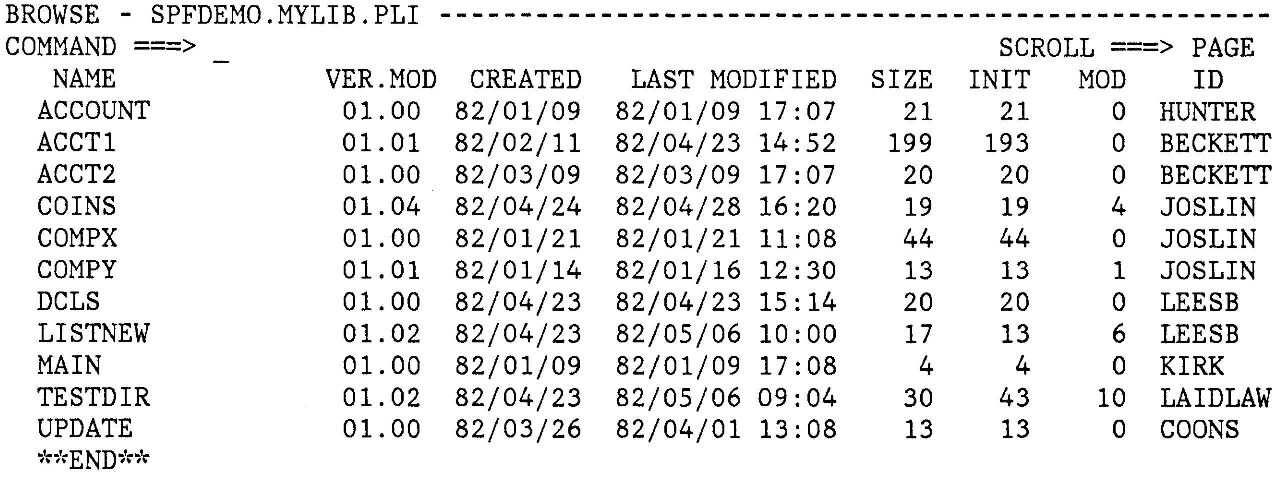
During browse and edit, the current version and modification level are displayed in the title area (line 1) following the library and member name.

On member lists, the following column headings are used:

NAME	- Name of the Member
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 (when member is first saved)
MOD	- Number of Modified Lines
ID	- User ID

The LIB column (library number) is displayed only if a concatenated sequence of libraries was specified. It indicates the library (1, 2, 3, or 4) in which the member was found.

Figure 11 shows an example of a member list with statistics.



```
BROWSE - SPFDEMO.MYLIB.PLI ---------------------------------------------
COMMAND ===> _                                          SCROLL ===> PAGE
   NAME             VER.MOD  CREATED    LAST MODIFIED   SIZE  INIT   MOD    ID
  ACCOUNT            01.00  82/01/09  82/01/09 17:07     21    21     0  HUNTER
  ACCT1              01.01  82/02/11  82/04/23 14:52    199   193     0  BECKETT
  ACCT2              01.00  82/03/09  82/03/09 17:07     20    20     0  BECKETT
  COINS              01.04  82/04/24  82/04/28 16:20     19    19     4  JOSLIN
  COMPX              01.00  82/01/21  82/01/21 11:08     44    44     0  JOSLIN
  COMPY              01.01  82/01/14  82/01/16 12:30     13    13     1  JOSLIN
  DCLS               01.00  82/04/23  82/04/23 15:14     20    20     0  LEESB
  LISTNEW            01.02  82/04/23  82/05/06 10:00     17    13     6  LEESB
  MAIN               01.00  82/01/09  82/01/09 17:08      4     4     0  KIRK
  TESTDIR            01.02  82/04/23  82/05/06 09:04     30    43    10  LAIDLAW
  UPDATE             01.00  82/03/26  82/04/01 13:08     13    13     0  COONS
  **END**
```

Figure 11. Member List Display

PARAMETER SPECIFICATION

Several entry panels require specification of an ISPF library or CMS file identifier, and (when applicable) a read or update password (or both). These items are described in the following paragraphs.

Library and File Identification

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

PDF panels prompt you for each component of the library identification as follows:

```
ISPF LIBRARY:
   PROJECT ===>
   LIBRARY ===>
   TYPE    ===>
   MEMBER  ===>          (blank for member selection list)
```

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

> **Note:** Before a library can be referenced, it must be defined to PDF with the specify library function of the file utility (option 3.2).

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

```
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 panels allow a concatenated sequence of up to four libraries. For example:

```
ISPF LIBRARY:
   PROJECT ===> SAMPLE
   LIBRARY ===> FRANK    ===> 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.FRANK.COBOL
SAMPLE.TEST.COBOL
SAMPLE.MASTER.COBOL
```

You can also specify CMS files on panels, as follows:

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

Enter the CMS file id according to normal CMS conventions; that is, filename, filetype, and (optionally) filemode, separated by one or more blanks.

If you specify both a library id and a CMS file id on the same panel, the CMS file id is used. Therefore, to specify a library, leave the CMS file id blank.

If the CMS file is on a minidisk which is not currently linked and accessed, PDF issues 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 you specify a filemode as part of the CMS file id, the specified filemode letter will be used in issuing the ACCESS command. Otherwise, the first available (currently unused) filemode letter will be used. Minidisks that are linked and accessed by PDF for you will remain linked and accessed only as long as they are being used by a given function (e.g., edit).

Passwords

PDF allows you to use VM password protection to permit read-only or read/write access to minidisks. You can specify passwords as follows:

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

Nondisplay input fields are used so that the passwords do not appear on the screen.

If the minidisk containing the library or CMS file is password protected, you 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 that cause data to be written to the minidisk (e.g., edit). In edit, if concatenated libraries are specified that 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) or multiple write (MW) password depending upon how the library was specified (see utility 3.2) or, for a CMS file, depending on what was specified on the panel for link access mode.

USER PROFILES

PDF "remembers" information on your behalf in your user profile. This allows PDF to prefill panel input fields with the values you last entered on the same (or a similar) panel. In some cases, default values are provided if you have not specified otherwise.

Information maintained in your user profile includes:

- 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 masks, tabs, and bounds
- Current scroll amount (browse, edit, and member lists)
- Processing options for each of the language processors
- File specification/information parameters

This information is automatically maintained from one session to another. As a new user, you will have to enter certain information the first time it appears. But from that point on, you simply verify the information and make whatever changes are desired before proceeding.

MEMBER LIST COMMANDS

For any of the following PDF options, a member list (Figure 11) is displayed if you specify an ISPF library or a CMS MACLIB or TXTLIB but supply no member name:

- Browse
- Edit
- Library Utility
- Move/Copy Utility
- Reset Statistics Utility
- Convert Menus/Messages Utility
- Foreground Processing
- Batch Processing

Locating Members

You can scroll the member list up and down by means of the scrolling commands. In addition, you can enter a LOCATE command in the command/option field on line 2 of any member list display. The format of the command is:

```
LOCATE  member-name
LOC
L
```

This causes a direct scroll to the specified member name (that is, the entry for the specified name appears 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 the browse, edit, foreground, and batch options, you can select one member at a time from the list, in either of two ways:

- Move the cursor down the left side of the screen and enter the single character "S" in front of the desired member name.

- Enter a SELECT command in the command field on line 2.

The format of the SELECT command is:

```
SELECT  member-name
SEL
S
```

This command also allows you to select a member that is not in the list. For edit, entering the SELECT command with the name of a nonexistent member is the mechanism for creating a new member with that member name.

When you terminate browse or edit, the member list is redisplayed with an automatic scroll to the member just processed. You can then select another member or enter the END command again to return to the browse or edit entry panel.

> **Note:** When the member list is redisplayed following browse or edit, it does not include new members that 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 browse or edit entry panel, leave the member name blank, and press ENTER.

Multiple-Selection Member Lists

For the utility options, you can select multiple members from the list by typing a single-character code in front of one or more member name(s) before you press ENTER.

S Select member (for move/copy, reset statistics, and convert utilities)

P Print member (for the library utility only)

R Rename member (for the library utility only)

D Delete member (for the library utility only)

B Browse member (for the library utility only)

When you press ENTER, the selected member(s) are processed and a confirmation indicator is displayed to the right of the member name. See the descriptions of the utilities for examples. You may then select additional members (scrolling to bring them into view, if necessary) or enter the END command to return to the previous panel.

LOG AND LIST FILES

PDF helps you obtain hardcopy listings of source modules, and also maintains a log of significant user activities. These items are kept in files called the list file and the log file, respectively.

When needed, these two files are generated automatically, and reside on your A-disk during the ISPF/PDF session. They are named:

```
  SPFLIST LISTING
  SPFLOG  LISTING
```

The list file is generated the first time you request a print function. The log file is generated the first time you perform some action that results in a log message, such as saving edited data or submitting a batch job. Once generated, these files remain open throughout your ISPF/PDF session.

The list file is used to accumulate PDF-produced listings and screen "snapshots". The list file accumulates, under user option, a listing of any source module that is created or modified by the PDF editor. Source listings and other types of printed output may also be obtained using the PDF utilities. A screen "snapshot" may be obtained by entering the PRINT or PRINT-HI command.

The log file maintains a log of significant user activities, including ISPF log messages, a record of files (and members) that were modified by edit or utility options, batch jobs that were submitted, dialog test trace data, etc.

Appendix C describes the format of the ISPF listing and log files.

You can print these files when you terminate ISPF. The Parms option (option 0.2) allows you to prevent generation of the ISPF log file (see Chapter 4). You can avoid generation of the ISPF list file by simply not requesting any print functions.

> **Note:** If you use the dialog test option, it is recommended that you allow generation of the log file.

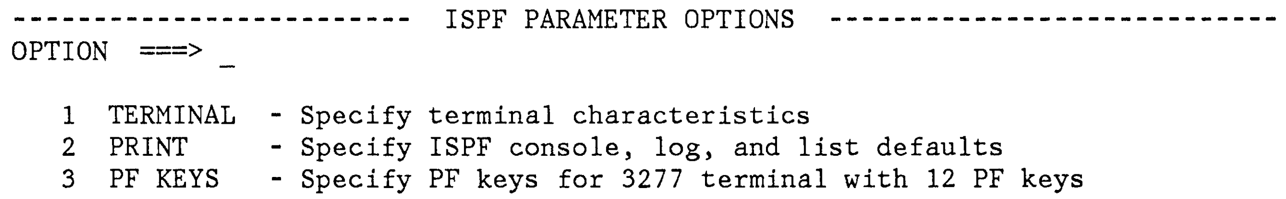
VIRTUAL CONSOLE

During PDF operation, all CMS commands and EXECs issued by PDF are spooled to your virtual console, including commands you request (under options 4 and 6) and commands that are automatically executed on your behalf (such as LINK and ACCESS commands).

Upon termination of ISPF, you may specify whether the virtual console is to be deleted or kept. If you specify that the console is to be deleted, PDF purges the console and sets it to "stop" status. If you specify that the console is to be kept, PDF leaves the console in "start" status. If the console remains in "start" status, it is closed (released for printing) by CMS when you log off.

CHAPTER 4. ISPF PARMS (OPTION 0)

The Parms option allows you to display and change a variety of ISPF parameters at any time during the ISPF/PDF session. Changes remain in effect until you change the parameter again, and are saved from session to session. The parameter selection panel is shown in Figure 12.

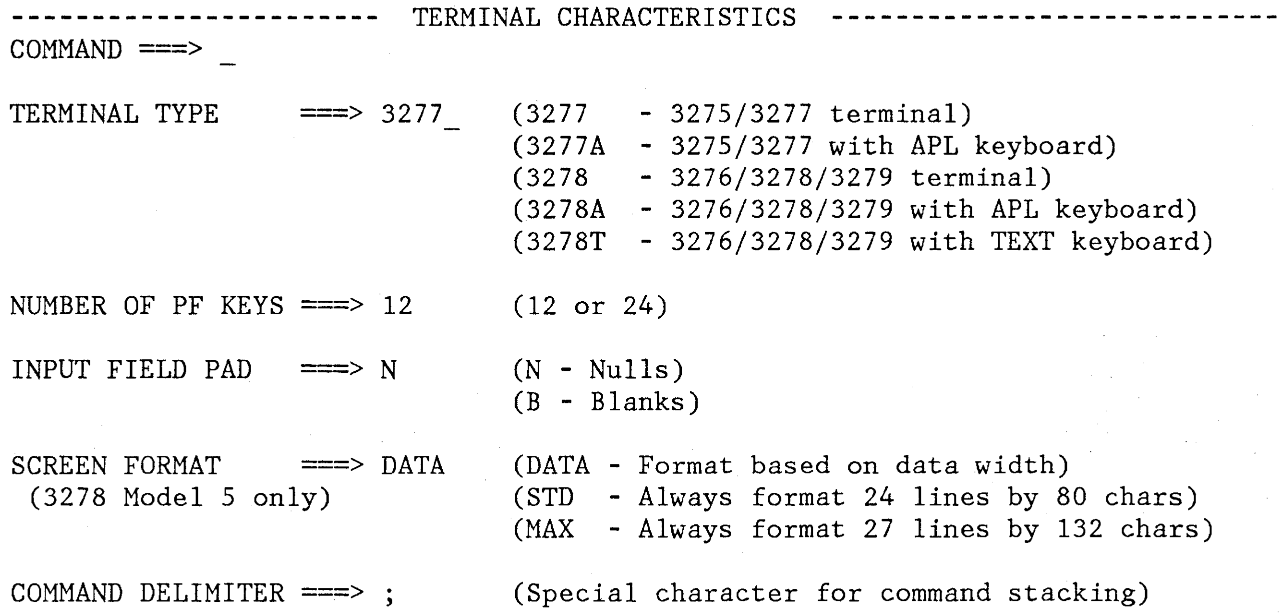


```
------------------------- ISPF PARAMETER OPTIONS ----------------------------
OPTION  ===> _

   1  TERMINAL  - Specify terminal characteristics
   2  PRINT     - Specify ISPF console, log, and list defaults
   3  PF KEYS   - Specify PF keys for 3277 terminal with 12 PF keys
```

Figure 12. Parameter Selection Panel

SPECIFY TERMINAL CHARACTERISTICS (OPTION 0.1)

When you select this option, a panel is displayed that allows you to specify the terminal type, number of program function (PF) keys, the default pad character for panel input fields, the mode of operation for a 3278 Model 5, and the command stacking delimiter. The panel, with the initial user defaults, is shown in Figure 13. (The allowable alternatives for these defaults are indicated on the display.)



```
------------------------  TERMINAL CHARACTERISTICS  ----------------------------
COMMAND ===> _

TERMINAL TYPE      ===> 3277_   (3277   - 3275/3277 terminal)
                                (3277A  - 3275/3277 with APL keyboard)
                                (3278   - 3276/3278/3279 terminal)
                                (3278A  - 3276/3278/3279 with APL keyboard)
                                (3278T  - 3276/3278/3279 with TEXT keyboard)

NUMBER OF PF KEYS ===> 12       (12 or 24)

INPUT FIELD PAD    ===> N       (N - Nulls)
                                (B - Blanks)

SCREEN FORMAT      ===> DATA    (DATA - Format based on data width)
 (3278 Model 5 only)            (STD  - Always format 24 lines by 80 chars)
                                (MAX  - Always format 27 lines by 132 chars)

COMMAND DELIMITER ===> ;        (Special character for command stacking)
```

Figure 13. Terminal Characteristics Panel

After you have reviewed these parameters, and changed them as necessary, enter the END command to return to the previous panel.

Specification of terminal type allows ISPF to recognize valid (displayable) characters. A 3278 terminal can display six more characters than a 3277. If you have a 3279 terminal, specify 3278 as the terminal type, since a 3279 terminal has the same character set as a 3278.

See Appendix D for the APL and TEXT character set descriptions.

> **Note:** One or more of following installation-dependent options for terminal type may also be included on this panel:
>
> 3278CF - for 3278 Canadian French terminals
> 3277KN - for 3277 Katakana terminals
> 3278KN - for 3278 Katakana terminals

Specification of the number of PF keys controls the particular set of PF key definitions currently in use, and also affects the panel displayed by option 0.3.

ISPF automatically sets (or changes) the number of PF keys:

- To 12, if the terminal type is 3277.
- To whatever was stored from your last ISPF session, if the terminal type is 3278. (For a new user, the number of keys is initialized to 12.)
- To 24, if you press a PF key higher than 12.

ISPF cannot sense the number of PF keys if you switch from a 3278 with 24 PF keys to a 3278 with 12 PF keys. In this case, you must inform ISPF of the number of PF keys using option 0.1 or 0.3. Otherwise, the incorrect set of stored key definitions will be used (see option 0.3).

Specification of a pad character controls the initial padding of panel input fields (including selection panels) but not the data portion of an edit display. Within edit, null or blank padding is controlled with edit commands.

Specification of screen format applies to 3278 Model 5 terminals only; it is ignored for other types of terminals. If you specify DATA, ISPF automatically uses the larger "default" mode characters (24 lines by 80 characters) or the smaller "native" mode characters (27 lines by 132 characters), depending on the width of the data to be displayed. If you specify STD or MAX, the mode is not switched.

You can stack commands on the command line of any panel by separating them with a delimiter. The default delimiter, the semicolon (;), may be changed using this option. Stacking allows you to enter, for example:

```
  ===> FIND DEPT;HEX ON
```

which finds the characters DEPT and then switches to hex mode.

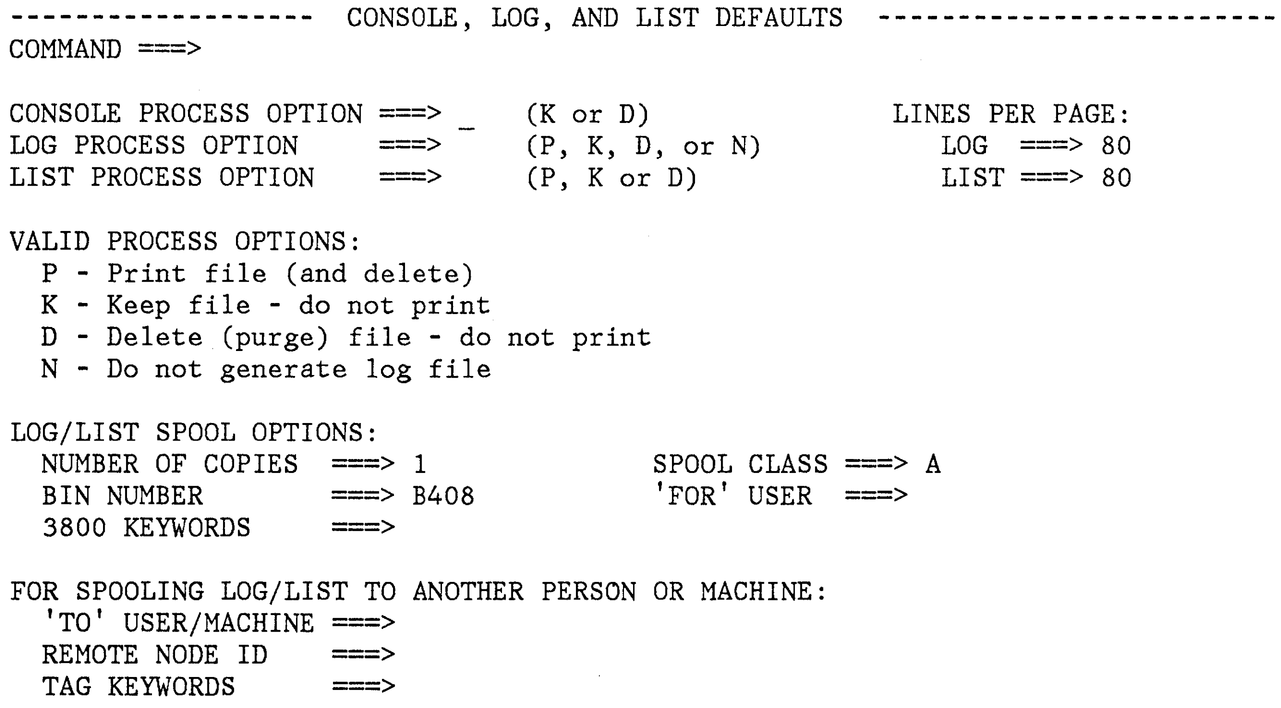
SPECIFY CONSOLE, LOG, AND LIST DEFAULTS (OPTION 0.2)

When you select this option, a panel (Figure 14) is displayed that allows you to specify default processing for the virtual console and for the log and list files. These entries will be used when you terminate ISPF using primary option X or the RETURN command. If you have not specified default processing options, primary option X will cause the termination panel to be displayed.

You may also specify the number of lines per page and spool parameters for printing the log and list files. The initial defaults are:

```
  Lines per page   ===> 80  (for both log and list)
  Number of copies ===> 1
  Spool class      ===> A
```

ISPF provides no defaults for the other parameters on this panel.



```
------------------- CONSOLE, LOG, AND LIST DEFAULTS -------------------------
COMMAND ===>

CONSOLE PROCESS OPTION ===> _   (K or D)                LINES PER PAGE:
LOG PROCESS OPTION     ===>     (P, K, D, or N)            LOG  ===> 80
LIST PROCESS OPTION    ===>     (P, K or D)                LIST ===> 80

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
  BIN NUMBER        ===> B408          '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 Panel

Normal values for lines per page are:

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

The virtual console is automatically started the first time ISPF invokes a CMS command (typically, a LINK or ACCESS executed automatically for you). Upon termination of ISPF, 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 you normally run with a virtual console

D - if you normally run without a virtual console

The ISPF log file is created the first time you perform some action that results in a log message, such as saving edited data or submitting a job to the batch machine. The ISPF list file is created the first time you request a print function.

For the log file, you may specify process option N to prevent generation of the log. You may avoid generation of the list file by simply not requesting any print functions.

> **Note:** If you use the dialog test option, it is recommended that you not prevent generation of the log file.

See the section entitled "PDF Termination" in Chapter 2 for a discussion of spool parameters that may be specified for the log and list files.

After you have reviewed the parameters on this panel, and changed them as necessary, enter the END command to return to the previous panel.

SPECIFY PROGRAM FUNCTION KEYS (OPTION 0.3)

The PF key definition panel allows you to assign PF keys to ISPF commands. You may assign PF keys to system commands (such as HELP or END), to commands that are meaningful within a particular function or environment (such as the edit FIND and CHANGE commands), and to line commands (such as edit or dialog test I or D commands).

When you enter the KEYS command or select option 3 from the ISPF Parms panel, the panel shown in Figure 15 is displayed.

The PF key definitions shown in the figure are the default definitions distributed with PDF.

Before you change your PF key assignments, you should verify the terminal type and the number of PF keys (12 or 24). The terminal type must be one of the following:

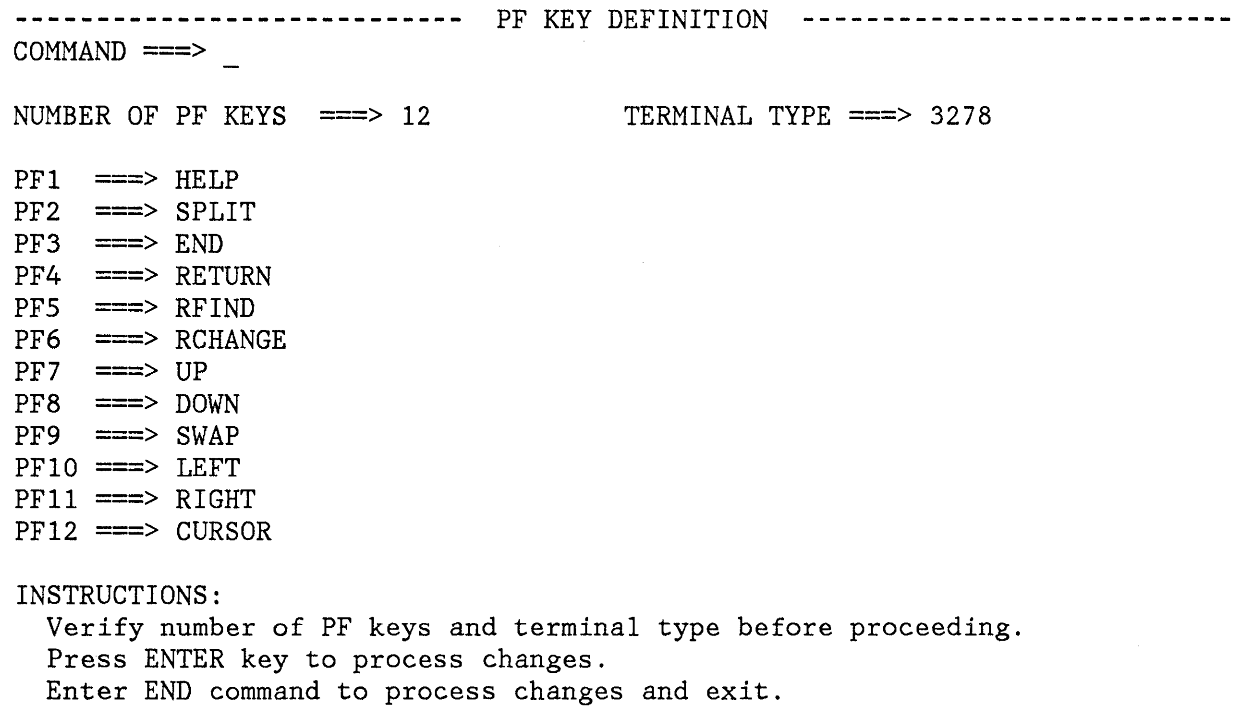
```
3277, 3277A, or 3277KN,
3278, 3278A, 3278T, 3278CF, or 3278KN
```

The panel shown in Figure 15 is the panel that is displayed for terminals with 12 PF keys. For terminals with 24 PF keys, the first panel displayed by the KEYS command or by option 0.3 shows the "primary" keys (PF13-PF24). When you press the ENTER key, a panel is displayed showing the "alternate" keys (PF1-PF12). You can flip-flop between the two panels by continuing to press ENTER. See Figure 16.

You can define or change a PF key function simply by equating the key to a command. For example:

```
PF9  ===> CHANGE ALL ABC XYZ
PF12 ===> PRINT
```

In the example, PF9 has been equated to an edit command, and PF12 has been equated to the system-defined PRINT command.



```
---------------------------  PF KEY DEFINITION  --------------------------
COMMAND ===> _

NUMBER OF PF KEYS  ===> 12           TERMINAL TYPE ===> 3278

PF1  ===> HELP
PF2  ===> SPLIT
PF3  ===> END
PF4  ===> RETURN
PF5  ===> RFIND
PF6  ===> RCHANGE
PF7  ===> UP
PF8  ===> DOWN
PF9  ===> SWAP
PF10 ===> LEFT
PF11 ===> RIGHT
PF12 ===> CURSOR

INSTRUCTIONS:
  Verify number of PF keys and terminal type before proceeding.
  Press ENTER key to process changes.
  Enter END command to process changes and exit.
```

Figure 15. PF Key Definition Panel (12 PF Keys)

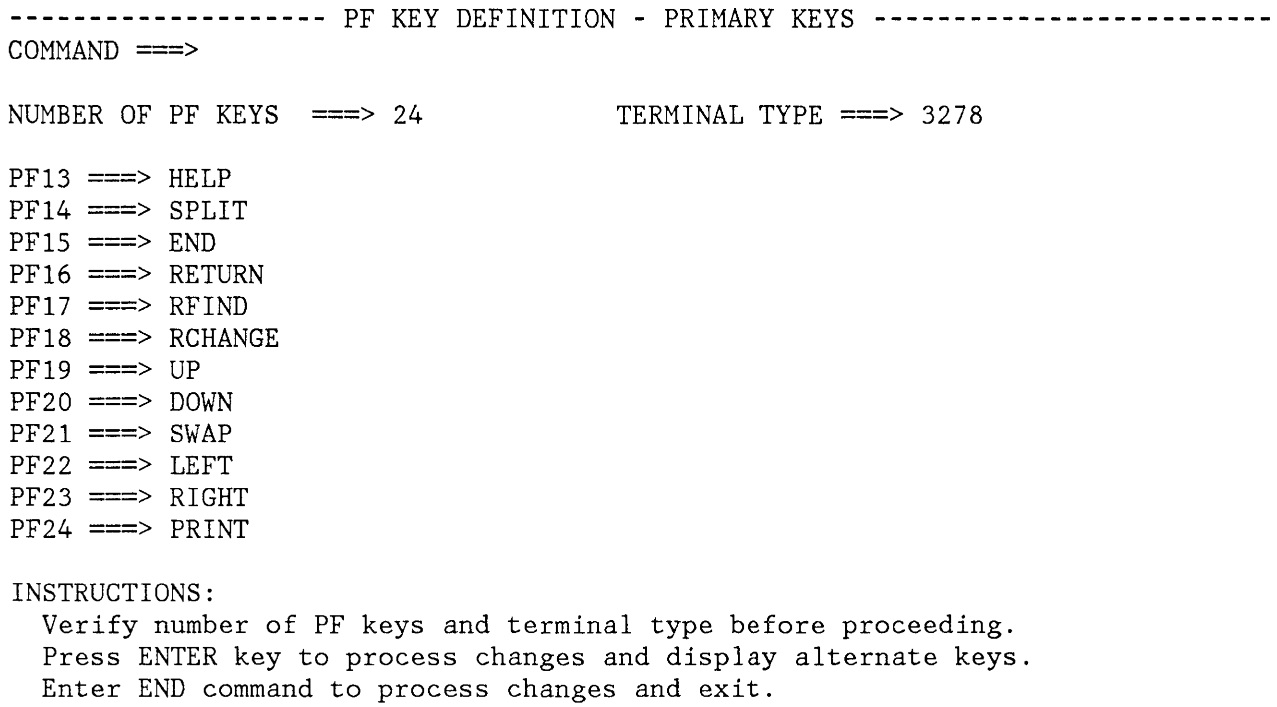
If you enter a blank for any PF key definition, the key is restored to its PDF default. The functions of the default PF key commands are discussed under "Program Function Keys."

When a PF key definition begins with a colon, it indicates a line command. The colon is stripped off and the command to which the key is equated is inserted in the first input field in the line at which the cursor is currently positioned. For example:

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

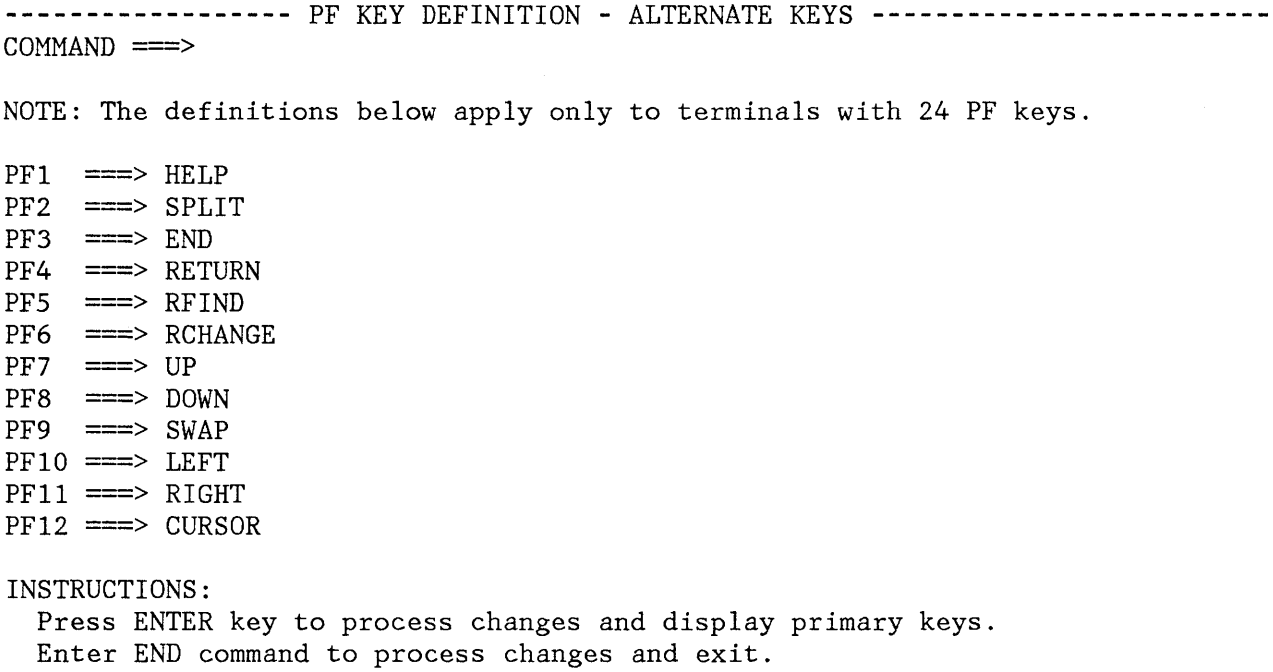
In this example, PF keys 10 and 11 have been equated to the edit text split (TS) and text flow (TF) line commands, respectively.

When a PF key definition begins with a greater-than sign, the command is assumed to be an edit or browse primary command, but not a system command. This is provided for compatibility with previous SPF products; the greater-than sign is no longer required.



```
-------------------- PF KEY DEFINITION - PRIMARY KEYS -------------------------
COMMAND ===>

NUMBER OF PF KEYS  ===> 24           TERMINAL TYPE ===> 3278

PF13 ===> HELP
PF14 ===> SPLIT
PF15 ===> END
PF16 ===> RETURN
PF17 ===> RFIND
PF18 ===> RCHANGE
PF19 ===> UP
PF20 ===> DOWN
PF21 ===> SWAP
PF22 ===> LEFT
PF23 ===> RIGHT
PF24 ===> PRINT

INSTRUCTIONS:
  Verify number of PF keys and terminal type before proceeding.
  Press ENTER key to process changes and display alternate keys.
  Enter END command to process changes and exit.
```

Figure 16 (Part 1 of 2). PF Key Definition Panels (24 PF Keys)



```
----------------- PF KEY DEFINITION - ALTERNATE KEYS ------------------------
COMMAND ===>

NOTE: The definitions below apply only to terminals with 24 PF keys.

PF1  ===> HELP
PF2  ===> SPLIT
PF3  ===> END
PF4  ===> RETURN
PF5  ===> RFIND
PF6  ===> RCHANGE
PF7  ===> UP
PF8  ===> DOWN
PF9  ===> SWAP
PF10 ===> LEFT
PF11 ===> RIGHT
PF12 ===> CURSOR

INSTRUCTIONS:
  Press ENTER key to process changes and display primary keys.
  Enter END command to process changes and exit.
```

Figure 16 (Part 2 of 2). PF Key Definition Panels (24 PF Keys)

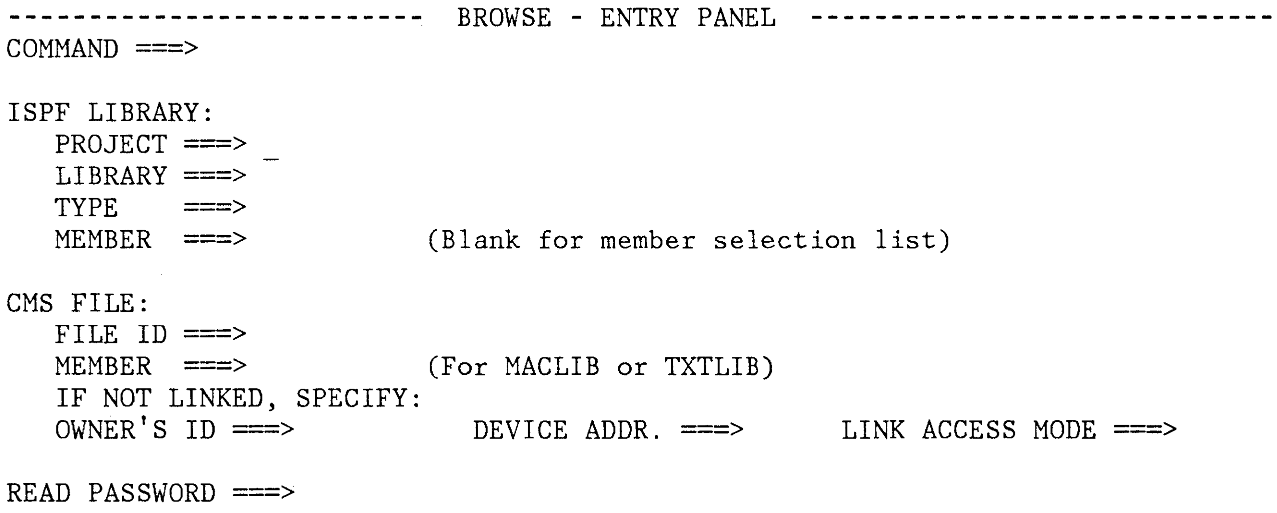
CHAPTER 5. BROWSE (OPTION 1)

The browse option allows you to display source data and listings stored in ISPF libraries or CMS files with the following characteristics:

- Fixed or variable record formats
- Logical record lengths up to and including 32,767 bytes

When you select browse, a panel (Figure 17) is displayed to allow you to specify either an ISPF library or a CMS file id. You can also specify a user id, virtual device address, LINK access mode, and read password, if needed, for linking to another user's virtual disk.

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

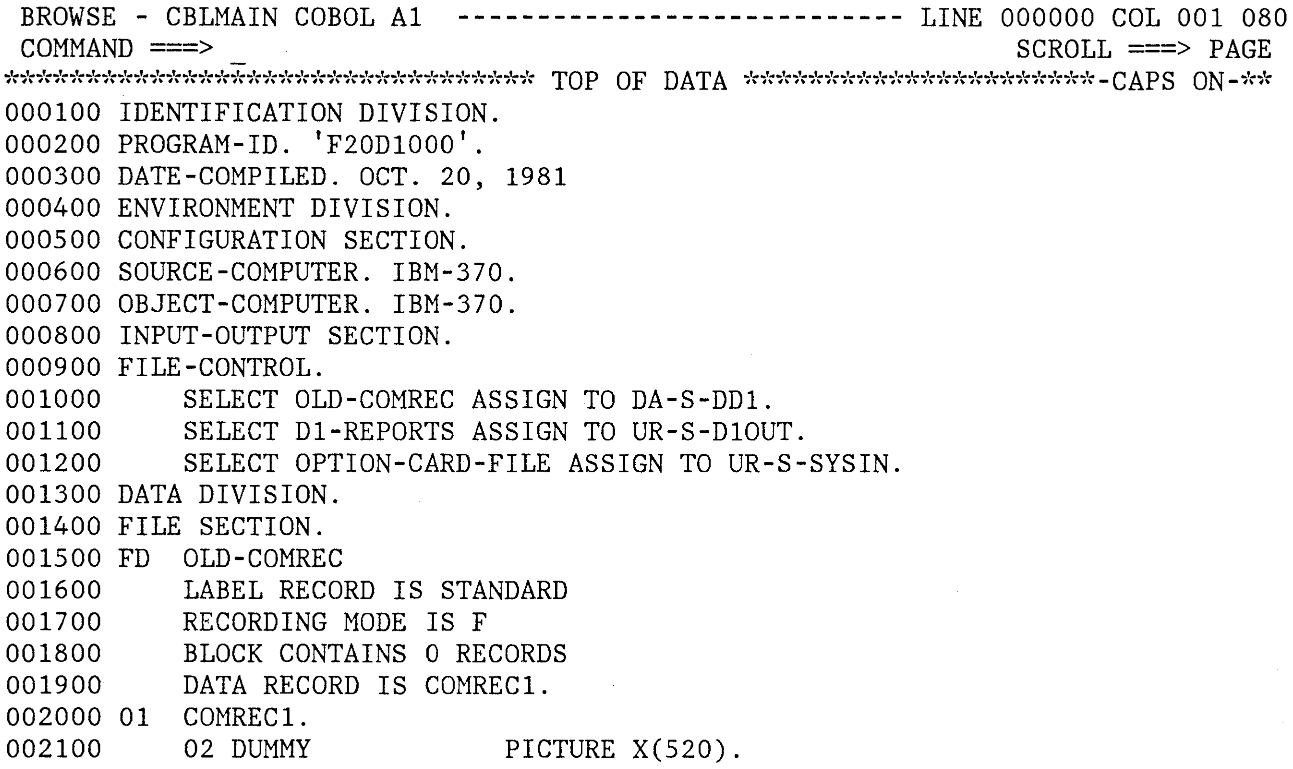


```
-------------------------  BROWSE - ENTRY PANEL  ---------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> _
   LIBRARY ===>
   TYPE    ===>
   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 17. Browse - Entry Panel

When you have properly specified the data to be viewed, the first page of data is displayed (Figure 18). Any invalid (nondisplayable) characters in the data are displayed as periods.

During browse, 4-way scrolling is available via the scrolling commands. You can also use the FIND and LOCATE commands to scroll to a particular character string, line number, or symbolic label.



```
 BROWSE - CBLMAIN COBOL A1  ---------------------------- LINE 000000 COL 001 080
 COMMAND ===> _                                                SCROLL ===> PAGE
********************************* TOP OF DATA **********************-CAPS ON-**
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. 'F20D1000'.
000300 DATE-COMPILED. OCT. 20, 1981
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).
```

Figure 18. Browse - Data Display

Browse provides six functions, described in the following sections, each of which is controlled by a command that you can enter in the command input field on line 2:

Locating Lines	(LOCATE)
Identifying Columns	(COLS)
Resetting Columns Line	(RESET)
Finding Character Strings	(FIND)
Setting Capitalization	(CAPS)
Controlling Hexadecimal Display	(HEX)

To terminate browse, enter the END command, which causes a return to the previous display (either the member list or the browse entry panel). When the return is to the member list, the member just browsed appears at the top of the list. You may select another member from the list or use END again to return to the browse entry panel.

When the entry panel is displayed, you may select another file or member, or you may use END to return to the primary option menu.

LOCATING LINES

Use the LOCATE command to bring a particular line to the top of the display. The line may be specified either by relative line number or by a previously defined label.

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

The LOCATE command has the following format:

```
LOCATE  line-number
LOC     label
L
```

You must enter either a line number or a label 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 line number relative to the beginning of the data. The line number is displayed in the upper right corner.

You can define a label by scrolling to any desired line and entering a label (on the command line) in the form:

```
.xxxxxxx
```

The label, a period followed by up to seven alphameric characters, is treated as an internal symbol and equated to the top line on the screen.

Once a label has been defined in this manner, it may be used as an operand in a subsequent LOCATE command. Labels are not retained when you leave the browse option.

> **Note:** The period is required 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

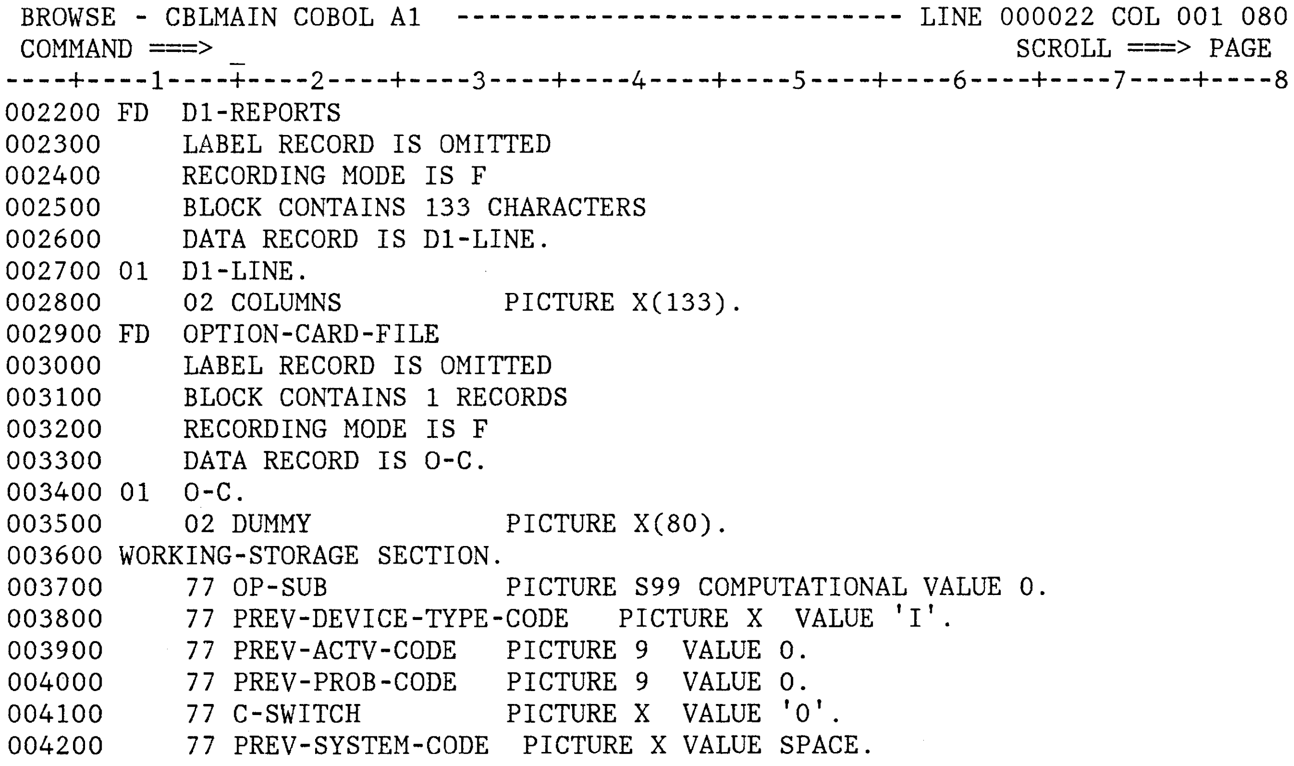
The COLUMNS and RESET commands are used to provide a temporary indication of where columns occur on the panel.

The COLUMNS command causes a column identification line to overlay the first line of data. The RESET command terminates display of the column identification line. The commands have the following formats:

```
COLUMNS
COLS
COL

RESET
RES
```

Neither command uses any parameters.

An example of the column identification line is shown in Figure 19. 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 000022 COL 001 080
 COMMAND ===> _                                                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.
```

Figure 19. Browse - Column Identification Line

FINDING CHARACTER STRINGS

To find one or more occurrences of a specified character string, use the FIND command. The FIND command allows special forms of character strings and several optional parameters to control the:

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

The command summary, which follows, describes the basic form of FIND that you will use most frequently. The full FIND command (as used with browse) is shown later in this section and is discussed in Chapter 6.

The basic format for FIND is:

```
FIND  string  [ALL]
F
```

The operands may be separated by blanks or commas. The string operand is the series of characters you wish to find. The ALL operand is optional, and may precede or follow the string.

Generally, you can type the string without delimiters. For example, enter:

```
FIND  XYZ
```

to find the next occurrence of XYZ.

You may delimit strings with either apostrophes (') or quotation marks ("). Use delimiters if a string contains imbedded blanks or commas, or if a string is the same as a command keyword. For example, enter:

```
FIND  'EVERY ONE'
```

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

If you omit the ALL operand, 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 that 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 corner of the screen. Use the RFIND PF key to find each successive occurrence of the string.

Note: If you type RFIND on the command line (rather than using the RFIND PF key), you must position the cursor to the desired starting location before pressing the ENTER key.

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 (if installed) is sounded. Then use the RFIND PF key to wrap to the top of data and continue searching. If the string is not found anywhere in the data, a "NO string FOUND" message is displayed.

If you type the ALL operand, the FIND command searches for all occurrences of the 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.

The complete format for FIND, showing all parameters used with browse, is:

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

The FIND command, as used with browse, has the same parameters as the FIND command used with edit. (The edit FIND command has one additional option that need not concern the browse user.) The full description of the FIND command is given in Chapter 6 in the section entitled "Finding and Changing Data."

TRANSLATING CHARACTER STRINGS

Translation of character strings entered in the FIND command is controlled by the CAPS command. If the character string operand of the FIND command contains alphabetic characters, they are automatically translated to uppercase if browse is operating with caps mode on; they are left as-is (not translated) if browse is operating with caps mode off.

You can turn caps mode on or off with the CAPS command. The format of the command is:

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

If you omit the ON/OFF operand, ON is assumed. Under browse, caps mode is always initialized ON.

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

DISPLAYING DATA IN HEXADECIMAL FORMAT

The display of data in hexadecimal notation is controlled by the HEX command. 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 (Figure 20).

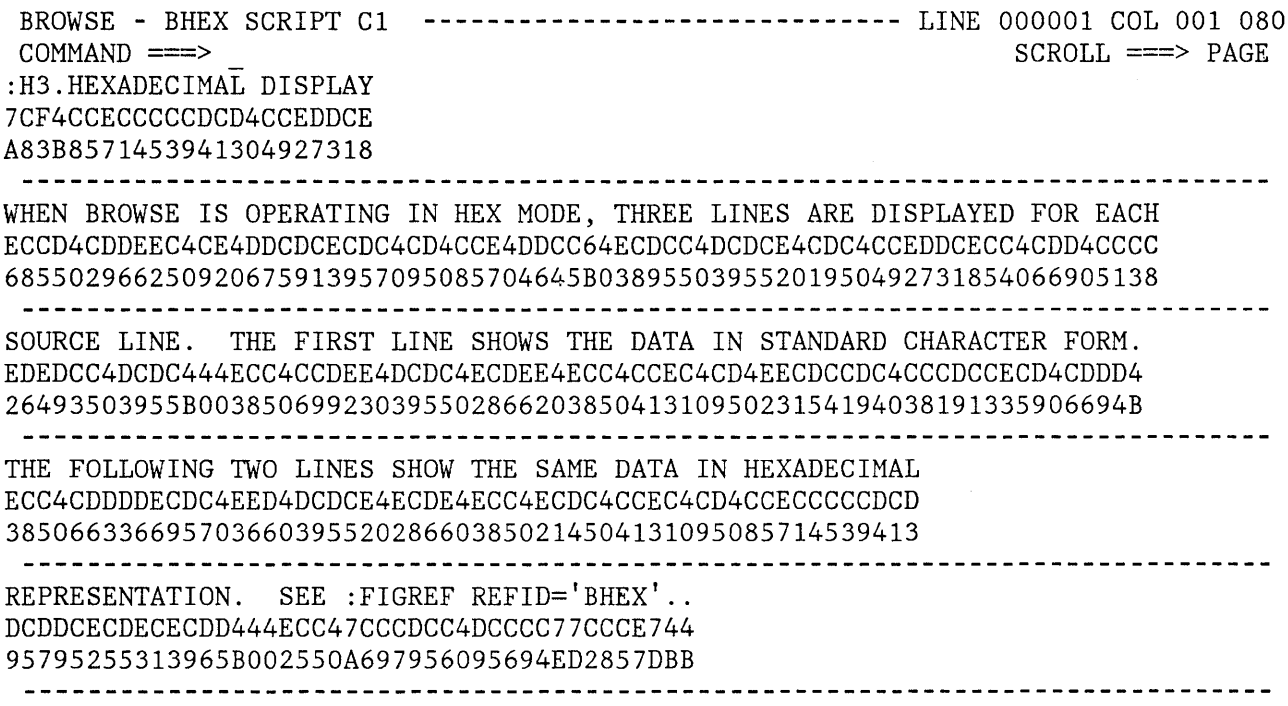
> **Note:** You may also use the FIND command to find invalid characters or any specific hex character, regardless of the setting of hex mode. See the discussion of picture strings and hex strings under the FIND command in Chapter 6.

You can turn hex mode on or off with the HEX command. The format of the command is:

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

The operands may be typed in any order. If you omit the ON/OFF operand, ON is assumed.

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 20. 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 you omit this operand, VERT is the default. A representation of the HEX DATA display is shown in the section on the HEX command under the edit option.



```
 BROWSE - BHEX SCRIPT C1  ----------------------------- LINE 000001 COL 001 080
 COMMAND ===> _                                               SCROLL ===> PAGE
:H3.HEXADECIMAL DISPLAY
7CF4CCECCCCCDCD4CCEDDCE
A83B85714539413049273 18
 ------------------------------------------------------------------------------
WHEN BROWSE IS OPERATING IN HEX MODE, THREE LINES ARE DISPLAYED FOR EACH
ECCD4CDDEEC4CE4DDCDCECDC4CD4CCE4DDCC64ECDCC4DCDCE4CDC4CCEDDCECC4CDD4CCCC
685502966250920675913957095085704645B038955039552019504927318540669051 38
 ------------------------------------------------------------------------------
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
38506633669570366039552028660385021450413109508571453941 3
 ------------------------------------------------------------------------------
REPRESENTATION.  SEE :FIGREF REFID='BHEX'..
DCDDCECDECECDD444ECC47CCCDCC4DCCCC77CCCE744
95795255313965B002550A697956095694ED2857DBB
 ------------------------------------------------------------------------------
```

Figure 20. Browse - Hexadecimal Display, Vertical

CHAPTER 6. EDIT (OPTION 2)

The edit option allows you to create, display, and modify source data (program code, test data, documentation, etc.) stored in ISPF 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

This chapter describes how you invoke and terminate edit, what functions you can perform with edit, and what commands you use to perform these functions.

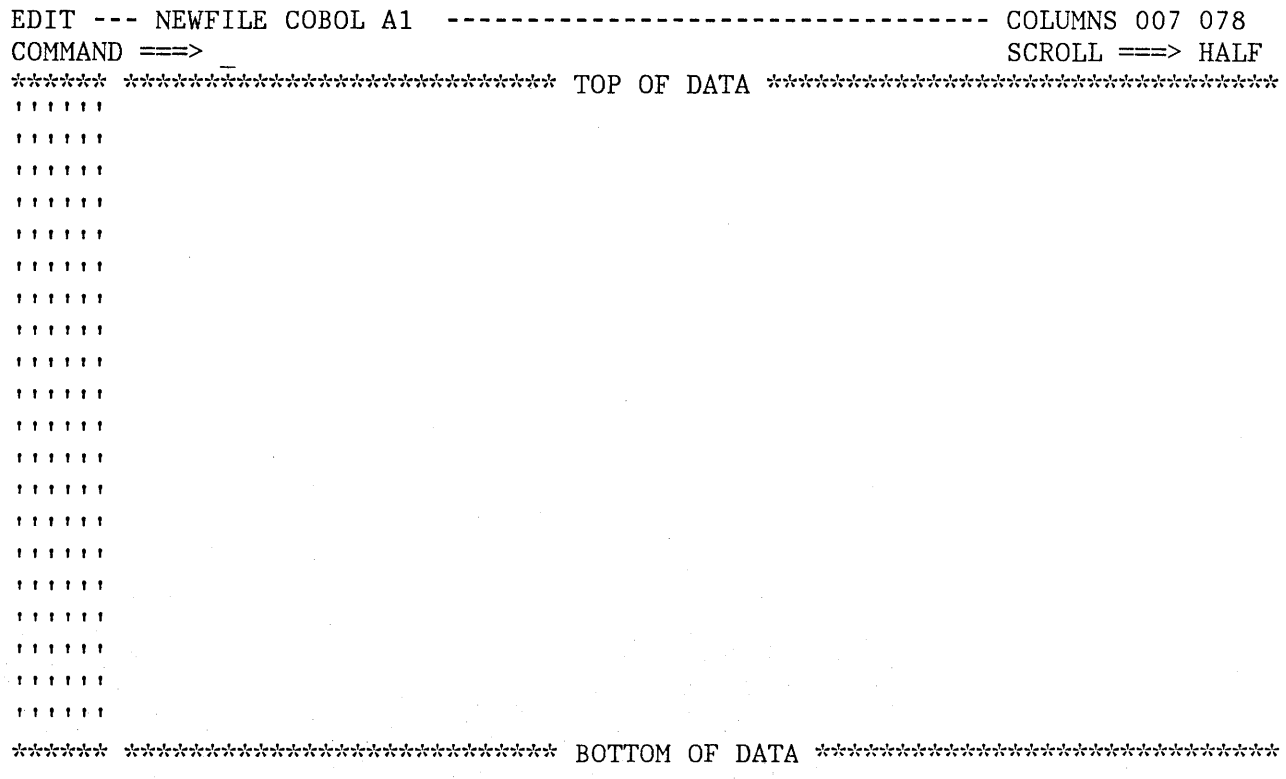
When using the edit option, keep in mind the following:

- Invalid (nondisplayable) characters are replaced on the screen (but not in the data) with attribute bytes, which are displayed as blanks and may not be overtyped. You may display and edit invalid characters by entering hex mode (see the description of the HEX command) or by using the FIND and CHANGE commands.
- The editor does not distinguish between "input mode" and "edit mode." Use the I (insert) line command to add new lines, either between existing lines or at the end of the data.
- When you specify an empty sequential file or nonexistent member of a library for editing, the initial display contains several inserted lines between the "TOP OF DATA" and "BOTTOM OF DATA" message lines (Figure 21).

EDIT SELECTION

The edit entry panel is shown in Figure 22. For edit, you may specify a concatenated sequence of ISPF libraries. The editor searches the libraries in the designated order to find the member and bring it into working storage. When you save the edited member, the editor places it (or replaces it) in the first library in the concatenation sequence regardless of which library it came from. Figure 22 indicates the entries you would make to display the member list for the concatenated libraries SPFDEMO.MYLIB.PLI and SPFDEMO.MASTER.PLI.

You must have previously specified the ISPF libraries to PDF. Refer to "File Utility (Option 3.2)" in Chapter 7 for details. Selecting a nonexistent CMS sequential file or a nonexistent member of a library allows you to create a new file or member.



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

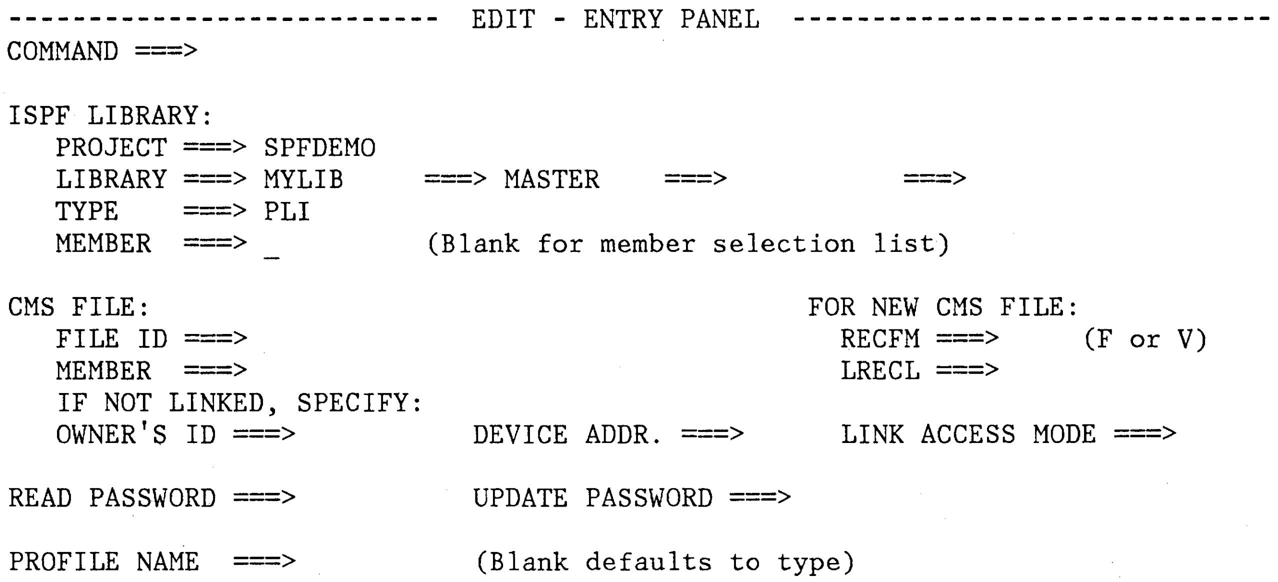
Figure 21. Edit - Creating New Data

If you request a member list, it includes an indication of the library where the member was found (library 1, 2, 3, or 4). See Figure 23.

If you specify a CMS file id and the file does not currently exist, you may specify the record format (RECFM) and logical record length (LRECL) on the panel. If you leave these fields blank, the RECFM and LRECL that you last specified for this filetype will be used. If you have never specified a RECFM and LRECL for this filetype, the editor creates the new file with RECFM=F (fixed) and LRECL=80.

If you specify a new member of an ISPF library (by specifying a member that does not exist), the RECFM and LRECL that you specified for the library will be used. See the description of the file utility (option 3.2).

If the minidisk is password protected, you must specify an update password. For a concatenated sequence of 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 accessible via the same password).



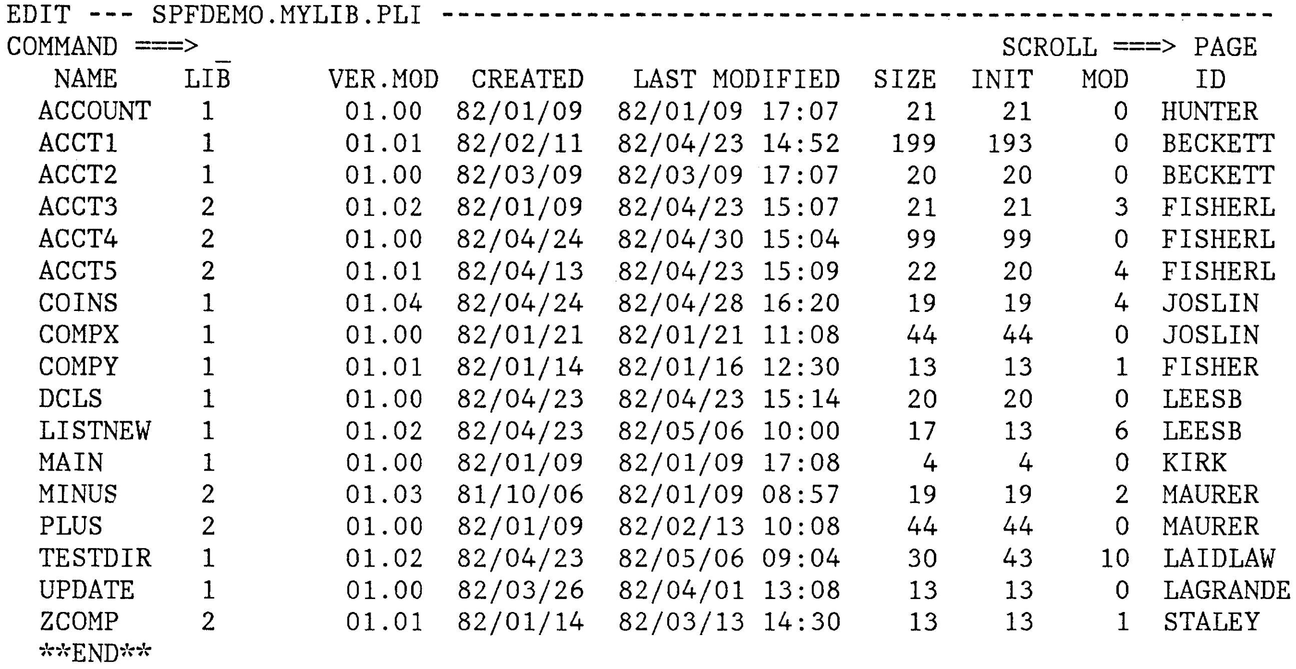
```
--------------------------   EDIT - ENTRY PANEL  ------------------------------
COMMAND ===>

ISPF 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 22. Edit - Entry Panel

The edit entry panel also allows you to specify a profile name, which you may enter to override the default edit profile. See the description under "Edit Profiles."

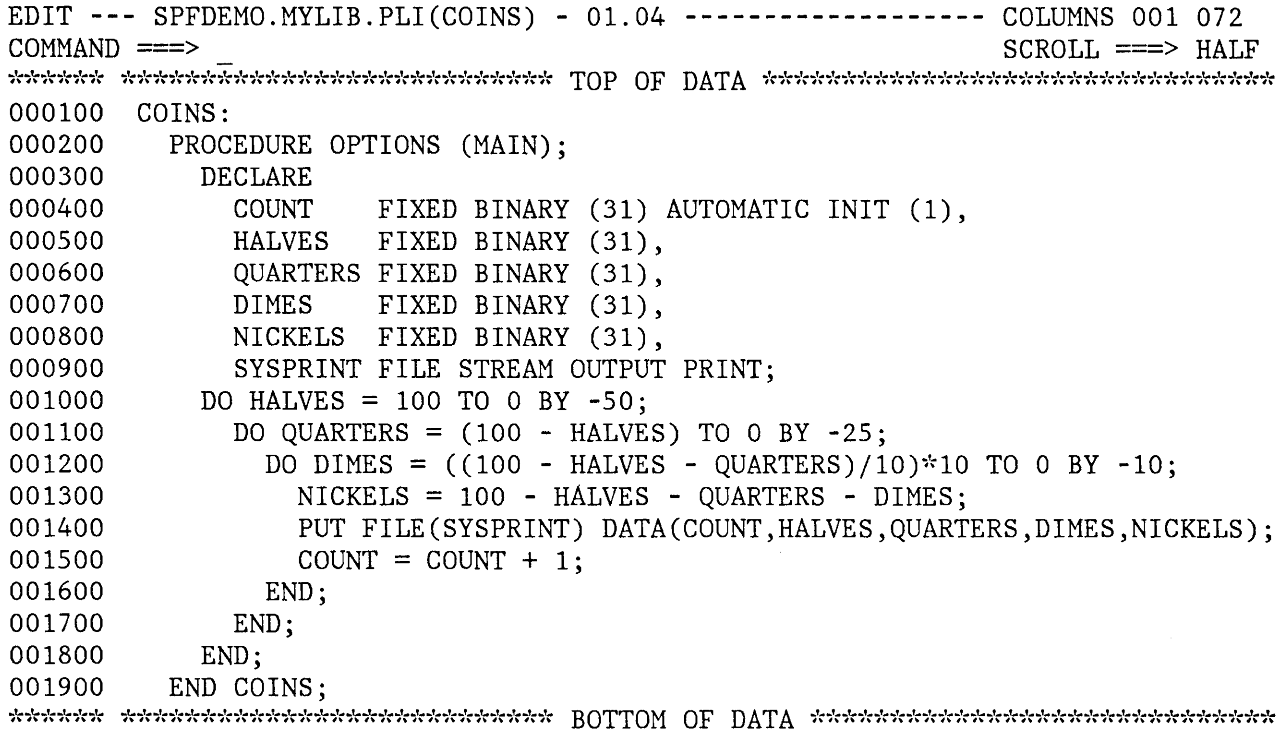
The selected member or sequential file is read into virtual storage, where it is updated during edit operations. Use of virtual storage for edit work space results in high performance, but may require a large virtual machine. If all available storage is consumed, an ABEND occurs, and you will lose the edit work space unless you operate with recovery mode on.

The edit data display is similar to the browse display except that each line consists of a 6-column line command field followed by a 72-column data field (Figure 24). The line command fields contain numbers that reflect the contents of the sequence numbers in the data. If the data has no sequence numbers, the line command fields contain numbers that start at 1 and are incremented by 1.



```
EDIT --- SPFDEMO.MYLIB.PLI -------------------------------------------------
COMMAND ===> _                                                SCROLL ===> PAGE
   NAME     LIB        VER.MOD  CREATED    LAST MODIFIED   SIZE  INIT   MOD    ID
  ACCOUNT    1          01.00  82/01/09   82/01/09 17:07    21    21     0  HUNTER
  ACCT1      1          01.01  82/02/11   82/04/23 14:52   199   193     0  BECKETT
  ACCT2      1          01.00  82/03/09   82/03/09 17:07    20    20     0  BECKETT
  ACCT3      2          01.02  82/01/09   82/04/23 15:07    21    21     3  FISHERL
  ACCT4      2          01.00  82/04/24   82/04/30 15:04    99    99     0  FISHERL
  ACCT5      2          01.01  82/04/13   82/04/23 15:09    22    20     4  FISHERL
  COINS      1          01.04  82/04/24   82/04/28 16:20    19    19     4  JOSLIN
  COMPX      1          01.00  82/01/21   82/01/21 11:08    44    44     0  JOSLIN
  COMPY      1          01.01  82/01/14   82/01/16 12:30    13    13     1  FISHER
  DCLS       1          01.00  82/04/23   82/04/23 15:14    20    20     0  LEESB
  LISTNEW    1          01.02  82/04/23   82/05/06 10:00    17    13     6  LEESB
  MAIN       1          01.00  82/01/09   82/01/09 17:08     4     4     0  KIRK
  MINUS      2          01.03  81/10/06   82/01/09 08:57    19    19     2  MAURER
  PLUS       2          01.00  82/01/09   82/02/13 10:08    44    44     0  MAURER
  TESTDIR    1          01.02  82/04/23   82/05/06 09:04    30    43    10  LAIDLAW
  UPDATE     1          01.00  82/03/26   82/04/01 13:08    13    13     0  LAGRANDE
  ZCOMP      2          01.01  82/01/14   82/03/13 14:30    13    13     1  STALEY
  **END**
```

Figure 23. Edit - Member List



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------ COLUMNS 001 072
COMMAND ===> _                                               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 24. Edit - Data Display

SUMMARY OF EDIT OPERATIONS

This section lists the functions that are performed by edit commands, and the modes and profiles in which the edit option may operate.

Command Functions

Two types of commands are used to control edit operations:

- Primary commands, entered in the command field at the top of the screen (line 2)

- Line commands, entered in the line command field on any line by overtyping the number displayed in that field

You can use primary commands to:

- Control your editing environment

- Locate a specific line
- Find and change a character string
- Merge and segment data (combine several members into one or split a member into two members)
- Submit data to the job stream
- Save the edited data or cancel without saving

You can use line commands to:

- Insert or delete lines
- Repeat lines
- Rearrange lines (moving or copying) or overlay portions of lines
- Prepare text
- Define a mask for input
- Change the indentation of data (shifting)
- Include or exclude lines from the display
- Control tabs and boundaries for editing

In addition, edit provides the MODEL command, which allows you to access a library of dialog element models with built-in tutorial information and to copy the models into the dialog you are editing. See the section entitled "Dialog Development Models."

A command summary may be found in Appendix E, and a quick-reference summary of the commands is contained in Appendix F.

Edit Modes

Edit can operate 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 library statistics.
- Autolist Mode - controls automatic recording of source listings in the ISPF list file.

- Caps Mode - controls translation of alphabetic characters to uppercase.

- Nulls Mode - controls replacement of trailing blanks with null characters on the display screen.

- Tabs Mode - controls use of "hardware" or "logical" tabs.

- Hex Mode - controls display of data in hexadecimal format.

- Recovery Mode - controls automatic recording of edit transactions to facilitate recovery following a system failure.

You can turn each mode on or off independently of the other modes with the following primary commands: NUMBER, AUTONUM, STATS, AUTOLIST, 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 that you can display at any time with the PROFILE command. When you terminate edit, the editor saves the current edit profile automatically as part of your user profile.

Edit Profiles

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

Each profile is normally associated with the data "type" (the ISPF library type or CMS filetype). When you edit COBOL data, for example, the default profile contains whatever modes were in effect the last time you finished editing COBOL data. You can create and use additional profiles by specifying a profile name on the edit entry panel, or as an operand on the PROFILE command. If you already have 25 edit profiles, the least recently used profile is automatically deleted when a new profile is generated.

Listed below are the default mode settings when a new profile is generated (i.e., when you edit a new type of data for the first time, or specify a new profile name on the edit entry panel):

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

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

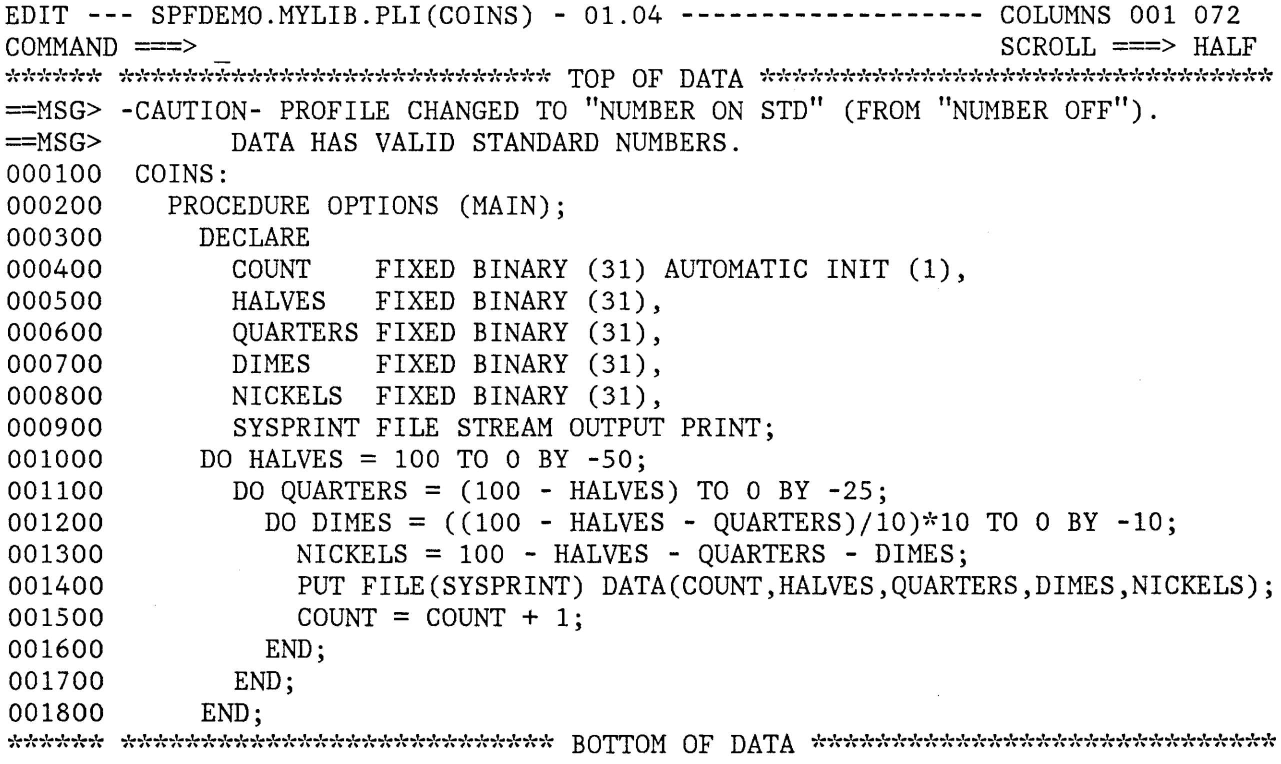
- Number mode is set on or off depending upon whether the data currently contains sequence numbers.

- Caps mode is set on or off depending upon whether the data currently contains lowercase alphabetics.

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

A caution message is displayed whenever edit automatically changes one of the mode settings. The message appears in the first two lines of the data area (Figure 25). A caution message is also displayed if edit discovers invalid (nondisplayable) characters in the data.

You can remove caution messages from the 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 ===> _                                                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;
****** ***************************** BOTTOM OF DATA ********************************
```

Figure 25. Edit - Caution Message

EDIT TERMINATION

Normally, you complete your edit session by entering the END command, 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 the description of the 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 entering a SAVE command or an END command.

   - The statistics are updated (or generated if none previously existed) if stats mode is on and the data is a member of an ISPF library.

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

2. A return is then made to the previous display (either the member list or the edit entry panel). When return is to the member list, the member just edited appears at the top of the list.

   Note: The RETURN command, which is logically equivalent to repeated use of the END command, also causes action 1 to occur.

You can save the data without terminating edit (and without printing) using the SAVE command. You can also terminate editing without saving (or printing) using the CANCEL command. Be aware that the CANCEL command cancels all changes made since the beginning of the edit session or the last SAVE command, whichever is more recent.

PRIMARY COMMANDS AND FUNCTIONS

This section deals with the commands that control your edit environment, allow you to find lines, change data, merge and segment data, and submit data for processing. All these commands are entered on line 2 of the panel.

Controlling Your Edit Environment

Your edit environment is that set of conditions in which you perform your PDF editing work.

Each condition is called a "mode" and is controlled by one or more commands. Generally, the name of the command and the mode name are the same. For example, the NUMBER command controls number mode, and determines whether your data lines are numbered.

The conditions that make up your edit environment are:

Sequence number generation and control (number mode)
Statistics generation (stats mode)
Source listing control (autolist mode)
Character translation (caps mode)
Nulls control (nulls mode)
Tab definition and control (tabs mode)
Hexadecimal display (hex mode)
Recovery control (recovery mode)

The majority of these modes have defaults that collectively provide an edit environment suitable for most users.

You can display the settings of these modes using the profile display and control facilities. You can also reset the display of information lines using the general reset facility.

Sequence Number Generation and Control

The following commands allow you to generate and control the numbering of lines (records) in your data:

NUMBER Turns number mode on or off

RENUM Renumbers all lines

UNNUM Turns off numbering and blanks the sequence number fields on all lines

AUTONUM Automatically renumbers data whenever it is saved

This section discusses the format of sequence numbers, how they are displayed, and how the number mode is initialized. Each of the four control commands is then discussed in detail.

Sequence Number Format

Sequence numbers normally start at 100 and are incremented by 100. When lines are inserted, the tens or units position are used. If necessary, one or more succeeding lines are renumbered automatically to keep the sequence numbers in order.

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 of the data (for fixed-length records) or the first eight characters of the data (for variable-length records), regardless of the programming language. The COBOL sequence field is always the first six characters of the data, and is valid only for fixed-length records.

For members of ISPF libraries, the format of standard sequence numbers is dependent on whether statistics are being generated. If statistics are being generated (i.e., stats mode is in effect) standard sequence numbers are six digits, followed by a 2-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 statistics, the 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 right justified within the field. COBOL sequence numbers are always six digits, and are unaffected by the setting of stats mode.

Sequence Number Display

For numbered data, the line command field (displayed at the left of each line) duplicates the sequence number in the data. Normally, the sequence numbers themselves are not displayed, but you may view them by scrolling left or right. You can include the sequence numbers in the data "window" by using the DISPLAY operand of the NUMBER or RENUM command.

When data is fetched for editing, the data is positioned on the screen so that the leftmost column of the data window is the first column displayed. When number mode is changed with a NUMBER, RENUM, or UNNUM command, automatic left or right scrolling is performed (if required) so that the leftmost column of the window is again the first column displayed.

For example, assume that the data has COBOL numbers in columns 1 through 6 and number mode (saved in the edit profile) is "NUMBER ON COBOL." When the data is fetched, column 7 will be the first column displayed. If you enter an UNNUM command, the data will be scrolled so that column 1 is the first column displayed. If you then enter a "NUMBER (or RENUM) COBOL" command, the data will be scrolled back to column 7. But if you enter a "NUMBER (or RENUM) COBOL DISPLAY" command, the data will be scrolled to column 1. (The sequence numbers in columns 1 through 6 will be part of the data window.)

Initialization of Number Mode

As data is fetched for editing, it is examined to determine whether it contains sequence numbers. The standard sequence field is always examined. The COBOL sequence field is also examined if the data "type" (ISPF 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 the number mode differs from the previous setting in the profile, a message is displayed indicating that edit has changed the mode. For new members or files, the initial setting of number mode is determined by the current profile. The default for a new profile is number mode on for standard sequence fields, and for COBOL fields if the data type is COBOL.

NUMBER Command

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

When operating in number mode, edit automatically generates sequence numbers in the data for new lines that are created when data is copied or inserted, and automatically renumbers the data when it is saved (provided autonum mode is also in effect).

The command has the following format:

```
NUMBER   [ON ] [STD] [COBOL] [DISPLAY]
NUM      [OFF]
```

You can enter the operands in any order. NUMBER ON is the default. The STD, COBOL, and DISPLAY operands are interpreted only when number mode is turned on. **COBOL** and **DISPLAY** may be abbreviated to **COB** and **DISPL** or **DIS**, respectively.

If you specify STD, the data is numbered in the standard sequence field. If you specify COBOL, the data is numbered in the COBOL field. If you specify both STD and COBOL, the data is numbered in both fields. If you omit both operands, the default is STD unless number mode was already in effect, in which case the data is numbered as previously specified.

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 that are out of sequence if the COBOL and STD fields are not synchronized. Use the RENUM command to force synchronization.

If you specify the DISPLAY operand, the width of the data window includes the sequence number fields. Otherwise, the width of the window excludes the sequence number fields. Automatic left or right scrolling is performed (if required) so that the leftmost column of the data window is the first column displayed.

RENUM Command

The RENUM command immediately turns on number mode and renumbers all lines, starting with number 100 and incrementing by 100. The command has the following format:

```
RENUM  [STD]  [COBOL]  [DISPLAY]
REN
```

The STD, COBOL, and DISPLAY operands are defined with the NUMBER command. **COBOL** and **DISPLAY** may be abbreviated to **COB** and **DISPL** or **DIS**, respectively.

UNNUM Command

The UNNUM command sets all sequence fields to blanks, turns off number mode, and positions the screen so that column 1 is the first column displayed. It is valid only when number mode is currently on. The standard sequence field, the COBOL sequence field, or both, are blanked, depending on what was being numbered prior to issuing the UNNUM command. The command has the following format:

```
UNNUM
UNN
```

The UNNUM command has no operands.

AUTONUM Command

The AUTONUM command controls autonum mode. When autonum mode is on, the data is automatically renumbered whenever it is saved, provided number mode is also on at that time. The command has the following format:

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

If you omit the operand, ON is assumed.

Statistics Generation (STATS Command)

The STATS command controls the generation of statistics for a member of an ISPF library. If stats mode is on when the member is saved, the statistics are 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. Stats mode is ignored for CMS files.

The command has the following format:

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

If you omit the operand, ON is assumed.

Whenever a member is fetched for editing, PDF checks the setting of stats mode. If stats mode is off and the member has statistics, PDF automatically turns on stats mode and displays a message indicating the mode change. If stats mode is on and the member has no statistics, a warning message is displayed, but the mode remains 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 statistics also affects the format of sequence numbers, as discussed in "Sequence Number Format."

Automatic Source Listing (AUTOLIST Command)

The AUTOLIST command controls the automatic generation of source listings. If autolist mode is on when you terminate edit, and if you have changed the data, the editor records a source listing of the data in the ISPF list file for eventual printing. The command has the following format:

```
  AUTOLIST  [ON ]
            [OFF]
```

If you omit the operand, ON is assumed.

> **Note:** This command replaces the edit PRINT command used in previous SPF products.

Character Translation (CAPS Command)

The CAPS command controls the caps mode. If caps mode is on, alphabetic data that you enter at the terminal is automatically translated to uppercase during edit operations. If caps mode is off, alphabetic data is left "as is." Caps mode is normally on for program development work and turned off 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 uppercase. Note that the keyword operands and all panel parameters are always translated to uppercase regardless of the current setting of caps mode.

The CAPS command has the following format:

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

If you omit the operand, ON is assumed.

Whenever data is fetched for editing, PDF initializes the setting of caps mode based on whether or not the data contains lowercase alphabetic characters. If the initial setting of caps mode differs from the setting in the profile, a message is displayed indicating that edit has changed the mode. For new members or files, the initial setting of caps mode is unchanged from the profile. The default for a new profile is caps mode on.

Nulls Control (NULLS Command)

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, you can split each line into multiple fields (see "Tabs Definition and Control").

You can turn nulls mode on or off with the NULLS command. The command has the following format:

```
  NULLS  [ON ] [ALL]
  NULL   [OFF]
```

You can type the operands in either order. If you omit the ON/OFF operand, ON is assumed.

The ALL operand is valid only when nulls mode is turned on. If you enter ALL, it causes all trailing blanks to be replaced with nulls and all-blank fields to be replaced with nulls. If you omit ALL, 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, you can create nulls at the end of a line using the ERASE EOF or DELETE key. Null characters are never stored in the data; they are always translated to blanks.

Tabs Definition (TABS Primary Command)

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 you press ENTER, the cursor is moved to the next software tab position (providing that you have defined software tabs).

Hardware tabs allow you to use the tab forward and tab backward keys to move the cursor to 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 you press the tab forward or tab backward key, the cursor is moved to the screen position immediately following the next or previous attribute byte. There are always at 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 you press the tab forward and tab backward keys. 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. See the discussion of the TABS primary command.

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

The three types of tabbing are controlled as follows:

- Hardware and logical tab positions are defined by means of the TABS line command (see description of line commands). Each asterisk (*) in the tabs line indicates the position of a hardware or logical tab. Hardware or logical tabs do not take effect, however, until tabs mode is turned on by the TABS primary command (see the description below).

- Software tab positions are also defined by means of the TABS line command. Each hyphen (-) or underscore (_) in the tabs line defines a software tab position or tab field. Software tabs take effect immediately upon being defined in the tabs line. They are unaffected by the setting of tabs mode and, hence, are unaffected by the use of the TABS primary command.

The TABS primary command controls the tabs mode and defines the logical tab character. The command has the following format:

```
  TABS  [ON ] [tab-character] [ALL]
  TAB   [OFF]
```

You can type the operands in any order. If you omit the ON/OFF operand, ON is assumed. 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; for example:

```
  TABS ON $
```

If you then enter the following information on a line:

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

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

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

If you do not enter the tab-character operand 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 you specify the ALL operand, an attribute byte is inserted at all user-defined positions on each line, overlaying any characters in those positions. If you omit the ALL operand, 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 takes place only on the display; the attribute bytes are never recorded in the data. When you turn tabs mode off, the attribute bytes are removed and the overlaid characters at each tab position are redisplayed.

In tabs mode, you can cause the attribute bytes to be temporarily removed from 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 ENTER. When you press ENTER again, the attribute bytes are reinserted.

Profile Display and Control (PROFILE Command)

You may display your current edit profile at any time by using the PROFILE primary command. You can also use this command to define a new profile or switch to a different profile. The command has the following format:

```
  PROFILE  [name] [number]
  PROF
  PRO
```

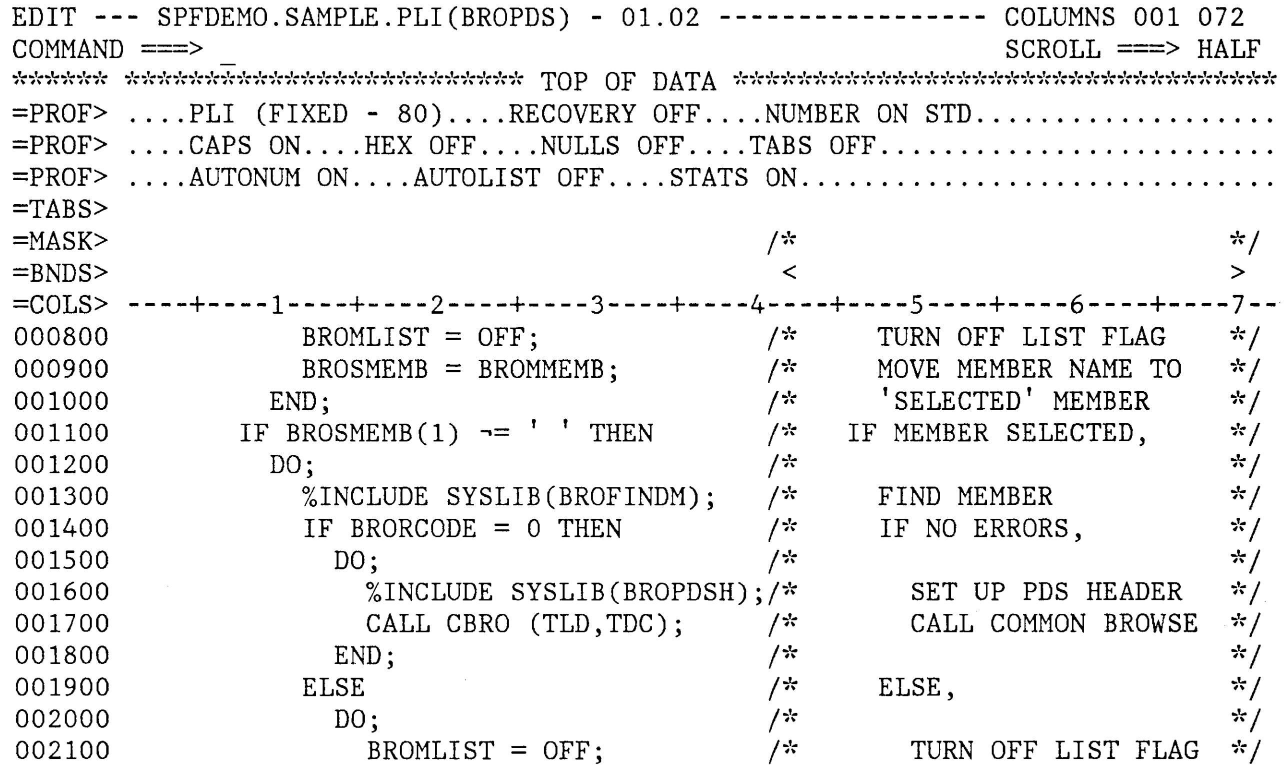
Both operands are optional, and may be entered in either order. The name operand consists of up to eight alphameric characters, the first of which must be alphabetic. The number operand consists of a single digit in the range 0 to 7, inclusive.

If you omit both operands, the contents of the profile are displayed in the data area of the screen (Figure 26). In the figure, the first three lines show the current mode settings. The remaining lines show the current contents of the TABS, MASK, and BOUNDS commands, together with the COLS positioning line. The MASK and TABS commands are not displayed if they contain all blanks, and the BOUNDS command is not displayed if it contains the default boundary positions. The remainder of the figure contains a portion of the data set being edited.

If you include the name operand, and if you already have a profile of that name, edit immediately switches to the specified profile and displays it. (You can avoid displaying the profile by typing a zero for the number operand.) 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 that was in effect when the PROFILE command was entered.

The number operand controls the number of lines shown in the profile display. If you type the number 0, the profile is not displayed. If you type a number from 1 through 6, that number of lines of the profile is displayed in the order shown in Figure 26. If you type the number 7, the complete profile is displayed, even if the mask and tabs are blanks and bounds contains the defaults. For the display in the figure, the user has entered "profile 7" to display all the lines, even the blank lines.

You can control the use of profiles from the edit entry panel. If you leave the profile name field blank, the profile name defaults to the data set type (last qualifier in the data set name). If you enter a name, 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, all-blank mask and tabs, and default bounds.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND ===> _                                                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....AUTOLIST OFF....STATS ON...............................
=TABS>
=MASK>                                         /*                           */
=BNDS>                                          <                           >
=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000800            BROMLIST = OFF;              /*     TURN OFF LIST FLAG    */
000900            BROSMEMB = BROMMEMB;         /*     MOVE MEMBER NAME TO   */
001000          END;                           /*     'SELECTED' MEMBER     */
001100        IF BROSMEMB(1) ¬= ' ' THEN       /*   IF MEMBER SELECTED,     */
001200          DO;                            /*                           */
001300            %INCLUDE SYSLIB(BROFINDM);   /*     FIND MEMBER           */
001400            IF BRORCODE = 0 THEN         /*     IF NO ERRORS,         */
001500              DO;                        /*                           */
001600                %INCLUDE SYSLIB(BROPDSH);/*       SET UP PDS HEADER   */
001700                CALL CBRO (TLD,TDC);     /*       CALL COMMON BROWSE  */
001800              END;                       /*                           */
001900            ELSE                         /*     ELSE,                 */
002000              DO;                        /*                           */
002100                BROMLIST = OFF;          /*       TURN OFF LIST FLAG  */
```

Figure 26. Edit - Profile Display

Hexadecimal Display (HEX Command)

When edit is operating in hexadecimal (hex) mode, three lines are displayed for each source line. The first line shows the data in standard character form. The next two lines show the same data in hexadecimal representation. See Figure 27.

You can change any invalid (nondisplayable) characters by overtyping the hexadecimal representations. You can change any valid character by overtyping either the standard character representations or the hexadecimal representations.

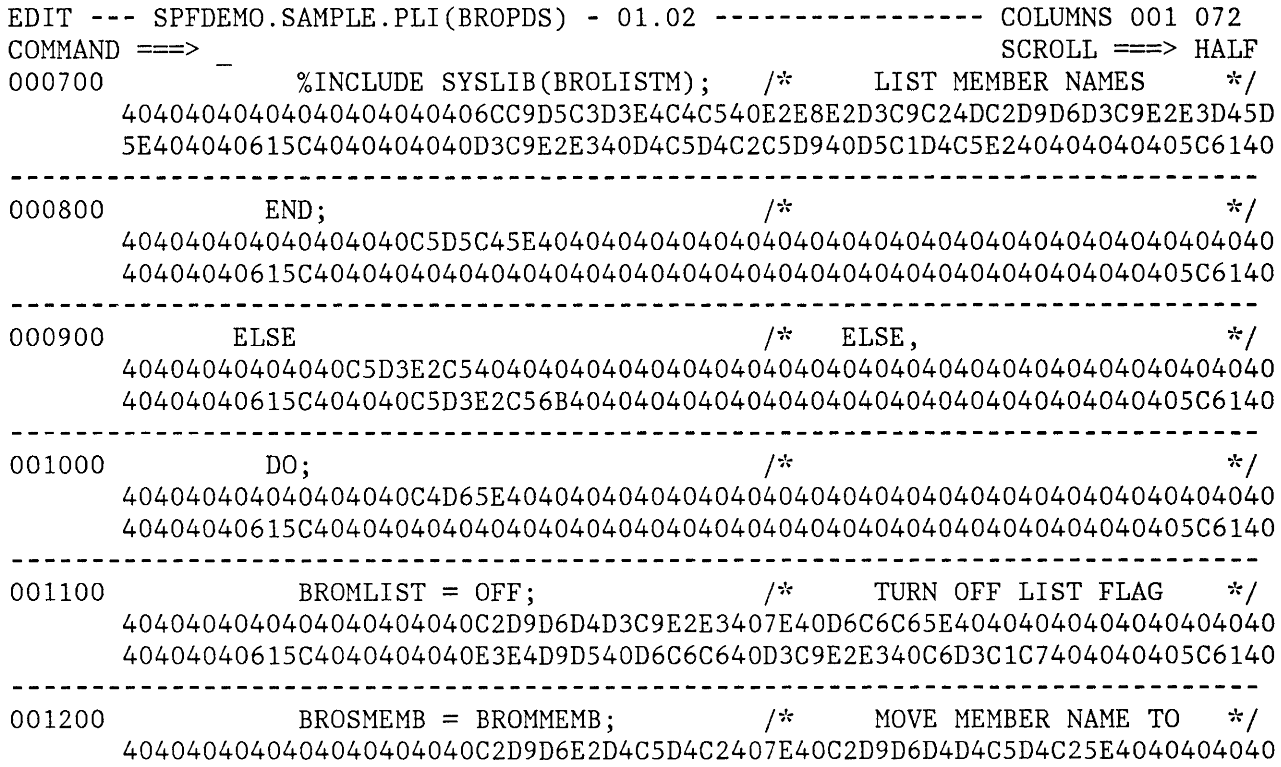
You can also use the FIND and CHANGE commands to find and change invalid characters or any specific hex character, regardless of the setting of hex mode. See the discussion of picture strings and hex strings under the FIND and CHANGE commands.

The HEX command controls hex mode. The command has the following format:

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

The operands may be typed in either order. If you omit the ON/OFF operand, ON is assumed.

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 27. Since the hex string is twice as long as the data string, it requires two rows. If you omit this operand, VERT is the default. A representation of the HEX VERT display is shown in Figure 20 on page 52.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ---------------- COLUMNS 001 072
COMMAND ===> _                                                 SCROLL ===> HALF
000700           %INCLUDE SYSLIB(BROLISTM);   /*     LIST MEMBER NAMES     */
       40404040404040404040406CC9D5C3D3E4C4C540E2E8E2D3C9C24DC2D9D6D3C9E2E3D45D
       5E404040615C4040404040D3C9E2E340D4C5D4C2C5D940D5C1D4C5E240404040405C6140
------------------------------------------------------------------------------
000800         END;                           /*                           */
       404040404040404040C5D5C45E4040404040404040404040404040404040404040404040
       40404040615C40404040404040404040404040404040404040404040404040405C6140
------------------------------------------------------------------------------
000900       ELSE                             /*   ELSE,                   */
       40404040404040C5D3E2C540404040404040404040404040404040404040404040404040
       40404040615C404040C5D3E2C56B404040404040404040404040404040404040405C6140
------------------------------------------------------------------------------
001000         DO;                            /*                           */
       4040404040404040C4D65E40404040404040404040404040404040404040404040404040
       40404040615C40404040404040404040404040404040404040404040404040405C6140
------------------------------------------------------------------------------
001100           BROMLIST = OFF;              /*     TURN OFF LIST FLAG    */
       40404040404040404040404040C2D9D6D4D3C9E2E3407E40D6C6C65E4040404040404040
       40404040615C4040404040E3E4D9D540D6C6C640D3C9E2E340C6D3C1C7404040405C6140
------------------------------------------------------------------------------
001200           BROSMEMB = BROMMEMB;         /*     MOVE MEMBER NAME TO   */
       4040404040404040404040C2D9D6E2D4C5D4C2407E40C2D9D6D4D4C5D4C25E4040404040
```

Figure 27. Edit - Hexadecimal Display, Data Representation

General Reset (RESET Command)

Edit displays warning messages in the data portion of the display if a mode setting has been automatically changed by edit, 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). You can reset the display of messages and other special conditions using the RESET command. The command has the following format:

```
RESET
RES
```

This command has no operands.

The RESET command terminates the display of all message lines and other special lines (COLS, MASK, TABS, and BOUNDS), resets line numbers to normal after they were replaced with "==CHG>", "==ERR>", or "=NOTE=" messages (refer to the CHANGE, SHIFT, and MODEL commands), blanks out pending commands, and redisplays all excluded lines. The RESET command does not change any edit modes, or alter 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.

Recovery Control (RECOVERY Command)

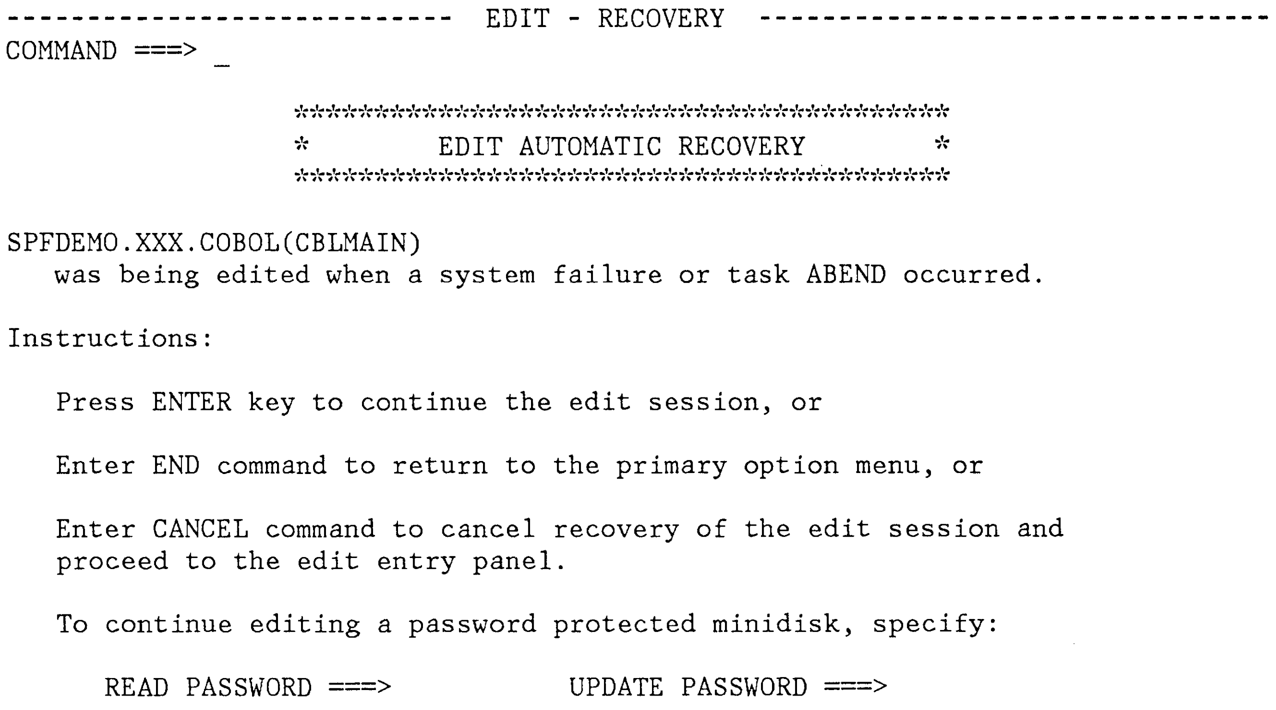
When edit is operating in recovery mode, an audit trail of user interactions is automatically recorded in a PDF-controlled temporary file. Following a system failure, you can use the audit trail to replay the edit session. When you select edit from the primary option menu, edit displays a special panel indicating that recovery is available (Figure 28). As directed on the panel, you can proceed with recovery, cancel recovery, or enter the END command to return to the primary option menu.

The RECOVERY command controls recovery mode. The command has the following format:

```
RECOVERY  [ON ]
REC       [OFF]
```

If you omit the operand, ON is assumed.

Operating with recovery mode off eliminates the I/O operations that maintain the recovery file and may therefore result in improved response time. 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 sequential file is being edited.



```
--------------------------- EDIT - RECOVERY -------------------------------
COMMAND ===> _

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

SPFDEMO.XXX.COBOL(CBLMAIN)
   was being edited when a system failure or task ABEND occurred.

Instructions:

   Press ENTER key to continue the edit session, or

   Enter END command to return to the primary option menu, or

   Enter CANCEL command to cancel recovery of the edit session and
   proceed to the edit entry panel.

   To continue editing a password protected minidisk, specify:

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

Figure 28. Edit - Recovery Panel

Locating Lines (LOCATE Command)

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 that precedes the specified line number. The command has the following format:

```
LOCATE  line-number
LOC
L
```

The line-number operand is a numeric value of up to eight digits (leading zeros need not be entered). If the operand contains six or fewer digits, it refers to the number in the line command field, displayed at the left of each line. (For numbered data, this field refers to the sequence numbers in the data records. For unnumbered data, these fields contain line numbers relative to the beginning of the data.) If the line-number operand contains eight digits, it refers directly to the sequence numbers in the data records, including the "modification flag" for ISPF libraries (see "Sequence Number Format").

Finding and Changing Data (FIND and CHANGE Commands)

The FIND and CHANGE commands, described in this section, allow you to find a specified character string and to change one character string to another. These are primary commands; you enter them in the command field on line 2 of the display rather than in the line command area.

Since these commands are often used together (to find a character string, and then change it to another string) they are discussed together. Both commands allow special forms of character strings and several optional parameters that control:

- The starting point and direction of search
- Special conditions that define a character string "match"
- Range and column limitations and the extent of the search

The following command summary 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 the FIND and CHANGE commands is shown below. Refer to the following section for a discussion of the complete commands.

```
FIND  string-1  [ALL]
F

CHANGE  string-1  string-2  [ALL]
CHG
C
```

If you omit the ALL operand, 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 that 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 corner of the screen. Use the RFIND or RCHANGE PF key to find or change each successive occurrence of the string.

> **Note:** If you type RFIND or RCHANGE on the command line (rather than using the PF keys), you must position the cursor to the desired starting location before pressing the ENTER key.

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 (if installed) is sounded. Use the RFIND or RCHANGE PF key to wrap to the top of data and continue searching. If the string is not found anywhere in the data, a "NO string-1 FOUND" message is displayed.

If you enter the ALL operand, 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.

You can separate the operands by blanks or commas. The ALL operand is optional, and may precede or follow the other operands.

Generally, you enter the strings without delimiters; for example, you might enter:

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

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

You must use delimiters if a string contains imbedded blanks or commas, or if a string is the same as a command keyword. You delimit strings with either apostrophes (') or quotation marks ("). For example, you might enter:

```
  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 two strings specified in a CHANGE command are not the same length, automatic shifting is performed by adding or deleting 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, you can reset the ==CHG> and ==ERR> line messages to normal by using the RESET command or by overtyping the line number or data on those lines.

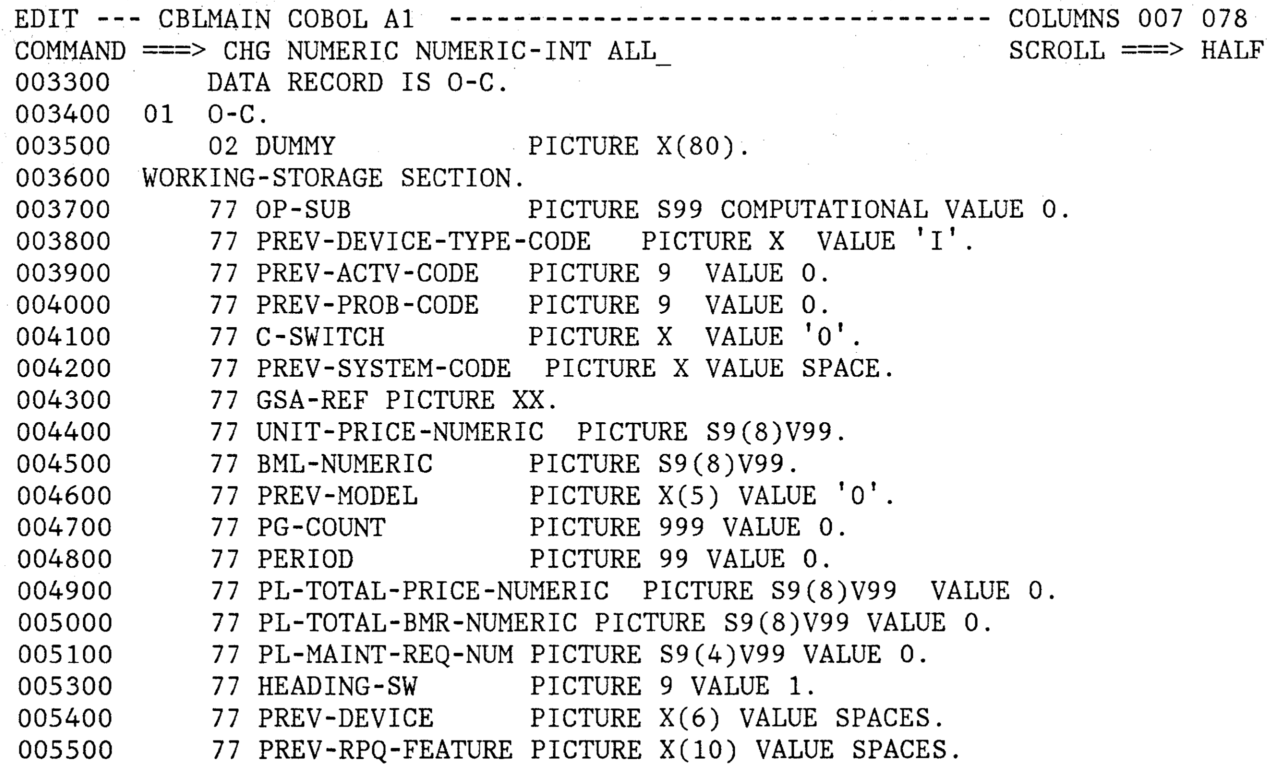
You can use the RFIND and RCHANGE PF keys together to find each occurrence of a character string, examine it, and then either change it (by pressing the RCHANGE PF key), or go on to the next occurrence (by pressing the RFIND PF key). To do this, type a CHANGE command on line 2 but then, rather than pressing ENTER, press the RFIND PF key. This will position the cursor to string-1 without changing it. Then press the RCHANGE PF key (to change it to string-2) or RFIND PF key (to get to the next occurrence of string-1).

You can also use the FIND and CHANGE commands with the X (exclude) line command to display only those lines that have been found or changed. For example, enter "X99999" in the line command area of the first displayed line to exclude all the lines from the display (see "Excluding Lines" in the section on "Line Commands and Functions"). Then enter a FIND or CHANGE command, such as:

```
FIND ALL XYZ
```

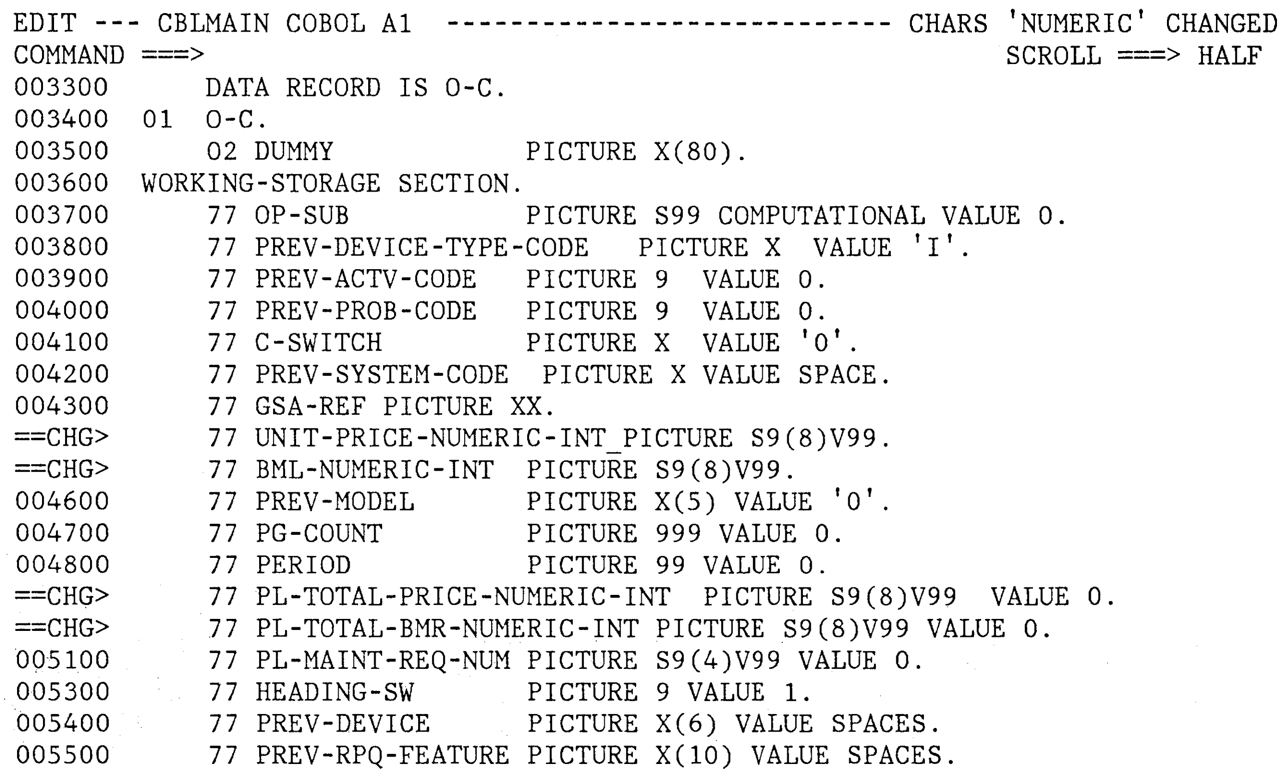
All lines containing the character string "XYZ" will "pop out" from the group of excluded lines.

Figure 29 shows a before-and-after example of the CHANGE command. 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 ===> 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.
```

Figure 29 (Part 1 of 2). Edit - CHANGE ALL Command Example



```
EDIT --- CBLMAIN COBOL A1  --------------------------- CHARS 'NUMERIC' CHANGED
COMMAND ===>                                                 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.
```

Figure 29 (Part 2 of 2). Edit - CHANGE ALL Command Example

Detailed Command Description

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

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

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

Note that FIND as an edit command (shown here) has the same format as FIND as a browse command, except for the addition of the optional X/NX parameter.

In both the FIND and CHANGE commands, PREFIX, SUFFIX, and CHARS may be abbreviated PRE, SUF, and CHAR, respectively.

You can separate the operands with blanks or commas, and type them in any order except that string-2 must follow string-1, and col-2 (if typed) must follow col-1. The string-1 operand is required, as is the string-2 operand for the CHANGE command; the others are optional.

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

- A simple string: Any series 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 by the character X. Example: X'C27B'

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

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

- A single asterisk (*). This causes the value that was used as string-1 in either a previous 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 that:

- You cannot specify a text or picture string for string-2.

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

The previous value of a character string (indicated by an asterisk or by use of the RFIND or RCHANGE PF keys) is retained until you end your editing session (i.e., until you return to the primary option menu).

Starting Point, Direction, and Extent of Search

You can control the starting point, direction, and extent of the search by using one of the following operands:

NEXT The scan starts at the current cursor location and proceeds forward to find the next occurrence of string-1. This is the default operand.

ALL The scan starts at the top of the data and proceeds forward to find all occurrences of string-1.

FIRST The scan starts at the top of the data and proceeds forward to find the first occurrence of string-1.

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

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

If the direction of the search is forward (i.e, if FIRST, ALL, or NEXT is specified), pressing the RFIND or RCHANGE 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 string-1 FOUND" message is displayed in the upper right 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, you can press the RFIND PF key (for either FIND or CHANGE) or the RCHANGE 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 string-1 FOUND" message is displayed.

Conditions for Character String Match

The operands CHARS, PREFIX, SUFFIX, and WORD control the conditions for a successful "match" with string-1 based on whether the data string begins and/or ends with a nonalphameric character (i.e., a special character or blank).

In the following illustration, the underscored strings would be found, and the non-underscored strings would be ignored.

```
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 no operand is specified, the default is CHARS.

Use of Text Strings

A text (T) string, which may be used as string-1 in a FIND or CHANGE command, allows the search to be satisfied without regard to uppercase or lowercase alphabetic characters. For 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.

Within the text string itself, alphabetics may be typed in uppercase or lowercase (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 user to search for a particular type of character, without regard for the specific character involved. You can use special characters 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 nonblank character
P'.'	-	any nondisplayable (invalid) character
P'#'	-	any numeric character (0-9)
P'-'	-	any non-numeric character
P'@'	-	any alphabetic character (upper or lowercase)
P'<'	-	any lowercase alphabetic character
P'>'	-	any uppercase alphabetic character
P'$'	-	any special character (not alphabetic or numeric)

If an APL or TEXT keyboard is being used, the following additional character is defined:

```
P'⌹' - any APL-specific or TEXT-specific character
```

Only the special characters listed above are valid within a picture string, but the string may include alphabetic or numeric characters, which represent themselves.

Examples of picture strings:

```
P'###'  - a string of three numeric characters
P'¬ ¬'  - any two nonblank characters separated by a blank
P'.'    - any nondisplayable 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 nondisplayable character
FIND P'¬' 72           - find next nonblank 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 nonblank
```

When you use the special characters '=' or '.', and a nondisplayable character is found, a hexadecimal representation is used in the confirmation message that appears in the upper right corner of the screen. For example:

```
FIND P'..'
```

could result in the message "CHARS X'02C5' FOUND".

Range and Column Limitations

You can limit the lines to be searched 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 that are excluded from the display.
NX   Scan only lines that are not excluded from the display.

If you omit this operand, both excluded and non-excluded lines are 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).

The col-1 and col-2 operands allow you to search only a portion of each line, rather than the complete line. These operands, integers separated by a comma or at least one blank, indicate the starting and ending columns for the search. If you specify col-1 without col-2, the string is found only if it starts in the specified column. If you specify neither col-1 nor col-2, the search continues across all columns within the current boundary columns (for edit - see the description of the BOUNDS command), or across all columns in the data (for browse).

Merging and Segmenting Data

You can merge or segment data 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, line commands are used to specify the source of the move or copy, and a primary command is used to specify the destination.

The MODEL command provides a specific copy facility, which allows you to copy predefined models of dialog parts with tutorial information into the data you are editing. For a description of the MODEL command, see the section entitled "Dialog Development Models."

Merging Data (COPY and MOVE Commands)

Use the COPY and MOVE primary commands (not be confused with the line commands of the same name) to specify a member of a library or a sequential file to be copied into the data being edited. If you use the MOVE command, the member or sequential file is deleted following a successful read operation. (For a concatenated sequence of ISPF libraries, the deletion occurs only if the member was in the first library of the concatenation sequence.) These commands are of the form:

```
COPY  [member-name]

MOVE  [member-name]
```

If you enter the member-name operand, the member is fetched from the ISPF library (or concatenated libraries) or the CMS MACLIB or TXTLIB being edited. The member-name operand is invalid if a CMS sequential file is being edited. If you omit the operand, a panel is displayed to allow you to specify any library member or file as the source for the COPY or MOVE.

You specify the destination for the COPY or MOVE with the A (after) or B (before) line command in the data being edited. You don't need to enter the A or B line command if the data being edited contains no lines (new member or empty file). A number may not follow the A or B line command when it is used with a COPY or MOVE primary command.

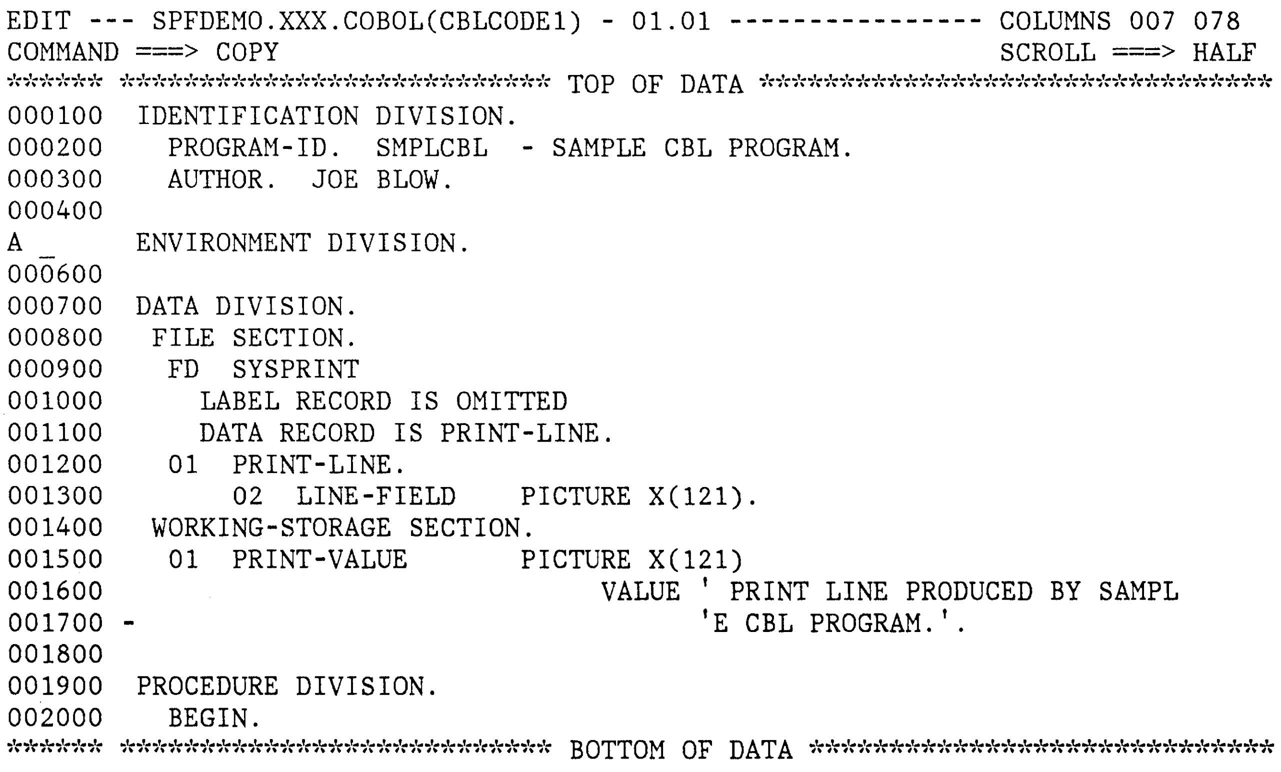
Note: When the member-name operand is omitted, the destination (A or B line command) must be specified before the COPY/MOVE panel is displayed.

The COPY or MOVE panel allows you to specify a range of lines in the source data. You must also specify the line searching criteria:

- 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.

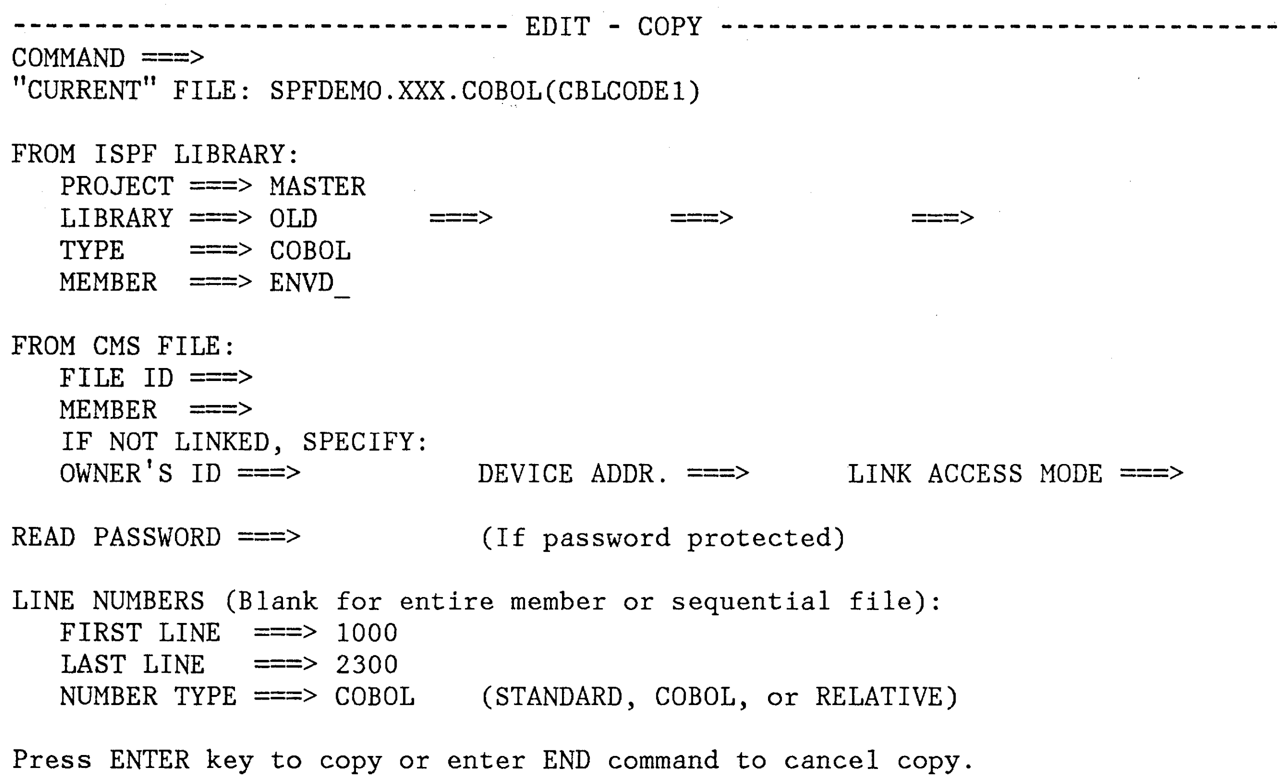
See "Sequence Number Generation and Control" for a further discussion of standard and COBOL line numbers.

Figure 30, Part 1, shows an example of the COPY command. Member CBLCODE1 is being edited in file 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 ENTER is pressed, the COPY panel is displayed. In Figure 30, Part 2, lines 1000 through 2300 of member ENVD in file MASTER.OLD.COBOL are to be copied. When the COPY panel has been filled in and ENTER is pressed, the lines are copied and inserted after line 500, and the edit data is redisplayed.



```
EDIT --- SPFDEMO.XXX.COBOL(CBLCODE1) - 01.01 --------------- COLUMNS 007 078
COMMAND ===> 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.
****** ***************************** BOTTOM OF DATA ****************************
```

Figure 30 (Part 1 of 2). Edit - COPY Command Example



```
------------------------------ EDIT - COPY ----------------------------------
COMMAND ===>
"CURRENT" FILE: SPFDEMO.XXX.COBOL(CBLCODE1)

FROM ISPF LIBRARY:
   PROJECT ===> MASTER
   LIBRARY ===> OLD       ===>          ===>          ===>
   TYPE    ===> COBOL
   MEMBER  ===> ENVD_

FROM 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 key to copy or enter END command to cancel copy.
```

Figure 30 (Part 2 of 2). Edit - COPY Command Example

Segmenting Data (CREATE and REPLACE Commands)

Use the CREATE and REPLACE primary commands to specify a file to be written from the data being edited. CREATE adds a member to a library or creates a new sequential file, providing a member or file of the same name does not already exist. REPLACE adds or replaces a member in a library or rewrites an entire file. The commands are of the form:

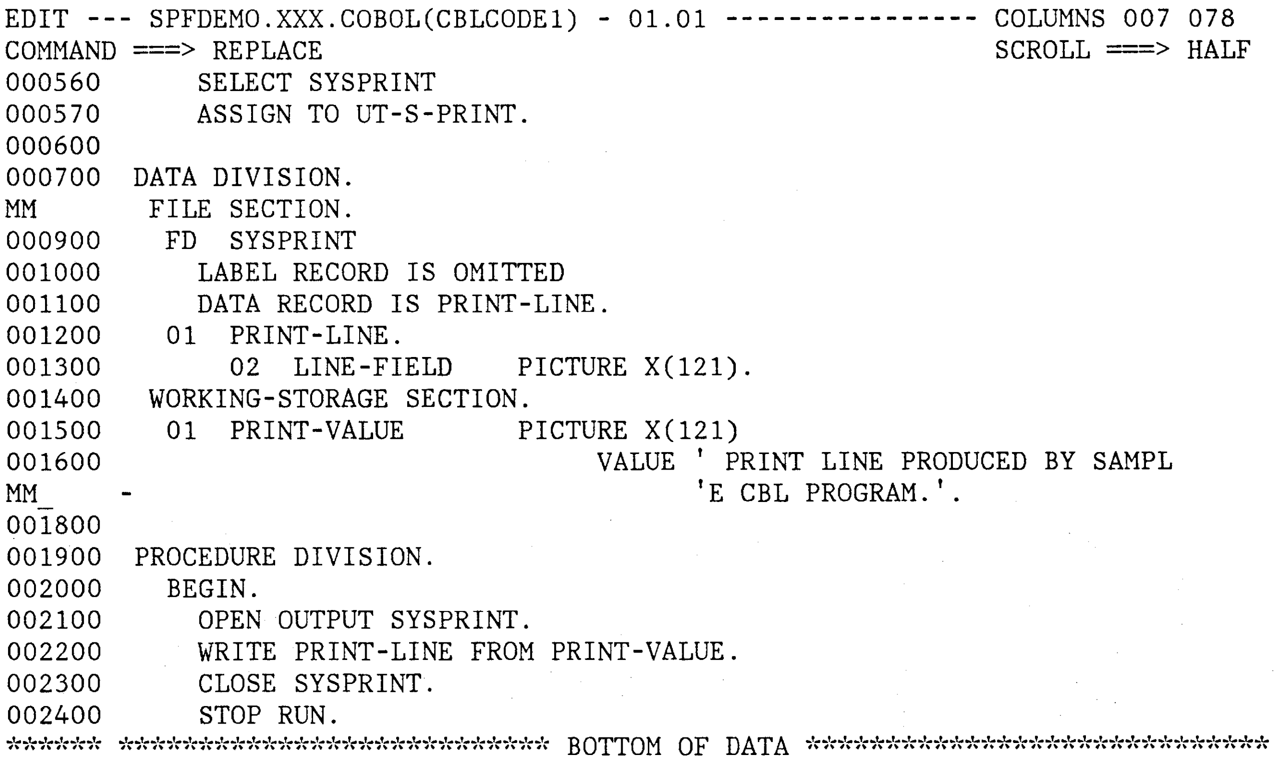
```
CREATE  [member-name]
CRE

REPLACE  [member-name]
REPL
REP
```

If you enter the member-name operand, the member is written in the ISPF library or CMS MACLIB or TXTLIB currently being edited. (For a concatenated sequence of ISPF libraries, the member is always written in the first library of the concatenation sequence.) The member-name operand is invalid if a CMS sequential file is being edited. If you omit the operand, you must specify the source with an M (move) or C (copy) line command and then a panel is displayed to allow you to specify any library member or file as the destination of the operation.

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

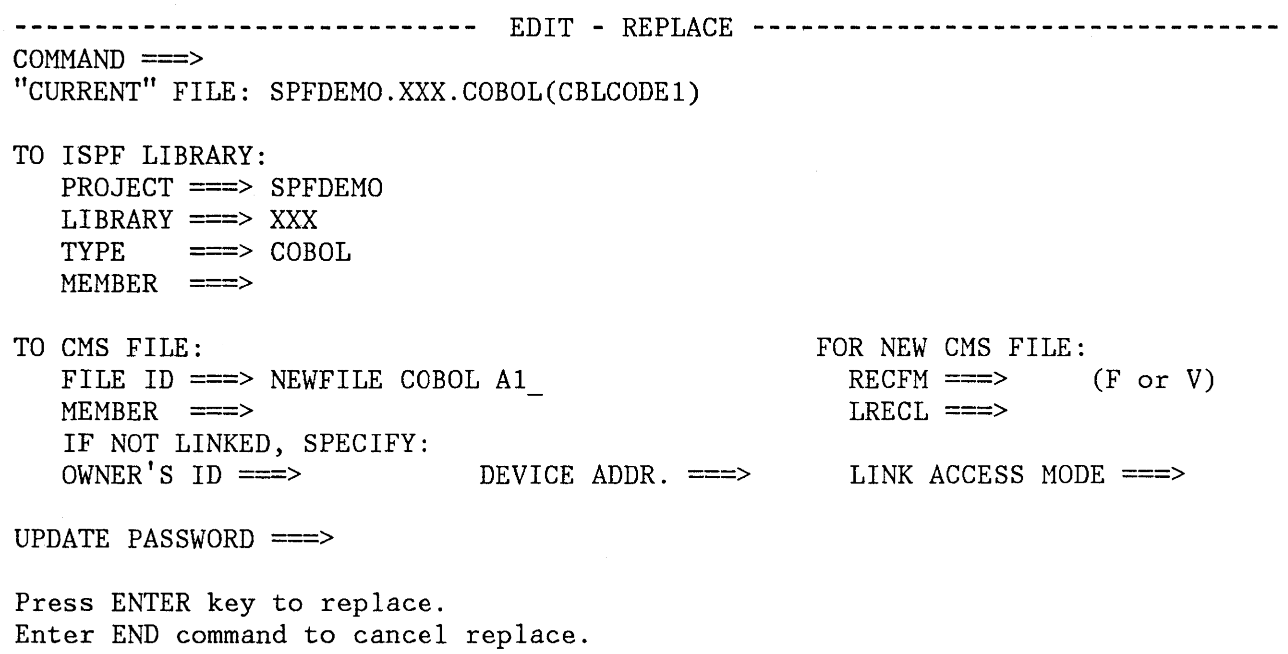
Figure 31 shows an example of the REPLACE command. Member CBLCODE1 is being edited in file SPFDEMO.XXX.COBOL. Lines 800 through 1700 are to be moved, as indicated by the "MM" commands. The entire contents of file NEWFILE COBOL A1 are to be replaced with lines 800 through 1700.

> **Note:** The data that is written to the destination file is renumbered if both number mode and autonum mode are in effect. A source listing of the data is also recorded in the ISPF list file for eventual printing, provided autolist mode is in effect.



```
EDIT --- SPFDEMO.XXX.COBOL(CBLCODE1) - 01.01 ---------------- COLUMNS 007 078
COMMAND ===> 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 ******************************
```

Figure 31 (Part 1 of 2). Edit - REPLACE Command Example



```
----------------------------  EDIT - REPLACE  ---------------------------------
COMMAND ===>
"CURRENT" FILE: SPFDEMO.XXX.COBOL(CBLCODE1)

TO ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX
   TYPE    ===> COBOL
   MEMBER  ===>

TO 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 key to replace.
Enter END command to cancel replace.
```

Figure 31 (Part 2 of 2). Edit - REPLACE Command Example

Submitting to a Job Stream (SUBMIT Command)

You can use the SUBMIT command to submit the data being edited (entire member or file) to the job stream of another VM machine. The command has the following format:

```
SUBMIT  machine-id
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 virtual machine.

Note: PDF does not supply a job statement when the edit SUBMIT command is invoked. You can supply job statements as part of the data being submitted.

Saving and Cancelling

These two commands allow you to preserve (save) the changes you've made without ending your edit session, and to end (cancel) your edit session without saving any changes you've made.

Saving Data (SAVE Command)

The SAVE command allows you to save edited data without ending your edit session. Generally, use of the SAVE command is not necessary if recovery mode is on. The command has the following format:

```
SAVE
```

This command has no operands.

The SAVE command writes the data back to the same file from which it was fetched, except when a concatenated sequence of ISPF 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 a library, the member is rewritten with the same member name, and the library statistics for the member are automatically updated (provided stats mode is in effect).

The data is automatically renumbered prior to saving, provided both number and autonum modes are in effect.

If SAVE cannot successfully rewrite the data due to I/O errors or insufficient space, a message is displayed in the upper right corner of the screen, accompanied by an audible alarm (if installed). You can 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 ENTER.

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

See the descriptions of the CREATE and REPLACE commands for further information.

Alternatively, you can enter split screen mode and attempt to fix the problem by using appropriate utilities, e.g., compress (for a library) or delete unused files from the minidisk, and then retry the save from the edit display.

You normally terminate edit by entering the END or RETURN command, 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 you enter the END or RETURN command.)

2. If any saves have occurred, the data is recorded in the ISPF list file (provided autolist mode is in effect).

3. Editing is then terminated.

Note: The SAVE command does not cause automatic recording in the ISPF list file, regardless of the setting of autolist mode.

Cancelling Changes (CANCEL Command)

The CANCEL command allows you to end your edit session without saving any of the changes you've made. This command is especially useful if you've made changes to the wrong data, or if the changes themselves are incorrect. If you have issued the SAVE command, and then issue the CANCEL command, the changes that you made before the SAVE command are not cancelled. The command has the following format:

```
CANCEL
CAN
```

This command has no operands.

The CANCEL command does not cause automatic recording in the ISPF list file, regardless of the setting of the autolist mode.

LINE COMMANDS AND FUNCTIONS

Line commands, used to edit data and text material, are entered by overtyping the 6-digit number in the line command area on one or more lines. The commands permit you to add or delete information, move, repeat, and copy individual lines or blocks of lines, change the indentation of lines, exclude or include lines in the display, and display a position-identification line that helps to align material in columns. In addition, a set of line commands (TE, TS, and TF) provide you with additional assistance in preparing text data.

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 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). You can type several line commands as well as data modifications before you press ENTER.

When you enter a line command or modify data and press the ENTER key, edit frequently repositions the cursor to another line number or to another location within the data. Edit attempts to anticipate the most appropriate position for the cursor, based on your previous action. To assist you in locating the cursor, edit intensifies the line number field on the line that contains the cursor. 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.

Rules for Entering Line Commands

The following rules apply to all edit line commands:

- You can enter several line commands, and make multiple data modifications, before you press the ENTER key. Error messages are displayed if the line commands are ambiguous. If you enter an erroneous or unwanted line command, you can undo it by simply retyping it, blanking it out, or entering the RESET command.

- Generally, you need to overtype only the first one or two characters of the line number to enter a line command. 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 you have not entered a number following the line command. If you want to repeat the line three times, you can 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 one or more blanks following the "R" but before the number:

     ```
     R 3700
     ```

  4. Type "R3" and press the ERASE EOF key to clear the rest of the line command field, or press ERASE EOF and then type "R3".

- You can enter the following line commands on the "TOP OF DATA" line by overtyping the asterisks that appear in its line command field:

I or In	Insert one or n lines ahead of the data
A or An	Move or copy one or n lines ahead of the data
TE or TEn	Enter one or n text lines ahead of the data

- You can enter the following line command on the "BOTTOM OF DATA" line by overtyping the asterisks:

B or Bn	Move or copy one or n lines following the data

Basic Line Commands

The following commands are described in this section:

I (Insert)
D (Delete)
R (Repeat)

M (Move)
C (Copy)
A (After)
B (Before)

I (Insert) Command

To insert a line, enter the single character "I" in the line command area, overtyping the line number. When you press the ENTER key, the editor inserts a new line 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, "I4" would cause four new lines to be inserted. The command has the following format:

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

If you enter any information 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 you press the ENTER key. If you enter no information on an inserted line, the line is automatically deleted the next time you press the ENTER key. If you enter information on the last (or only) inserted line and if the cursor is still in the data portion of that line when you press ENTER, another new line is automatically inserted following that line. This allows line after line of data to be generated in a "continuous insert" mode.

Figure 32 shows a before-and-after example of line insertion, in which three new lines are inserted after line 800.

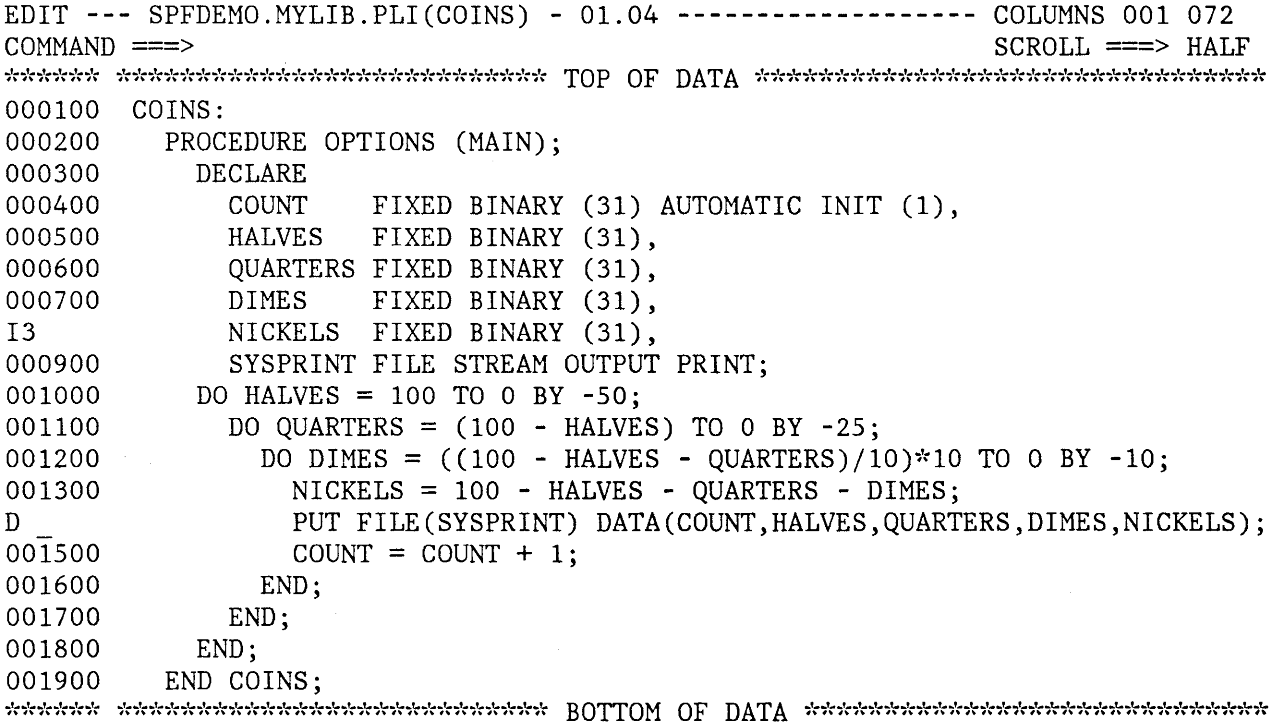
D (Delete) Command

To delete a line, enter the single character "D" 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" causes all remaining lines to be deleted, starting with the line containing the "D99999". The command has the following format:

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

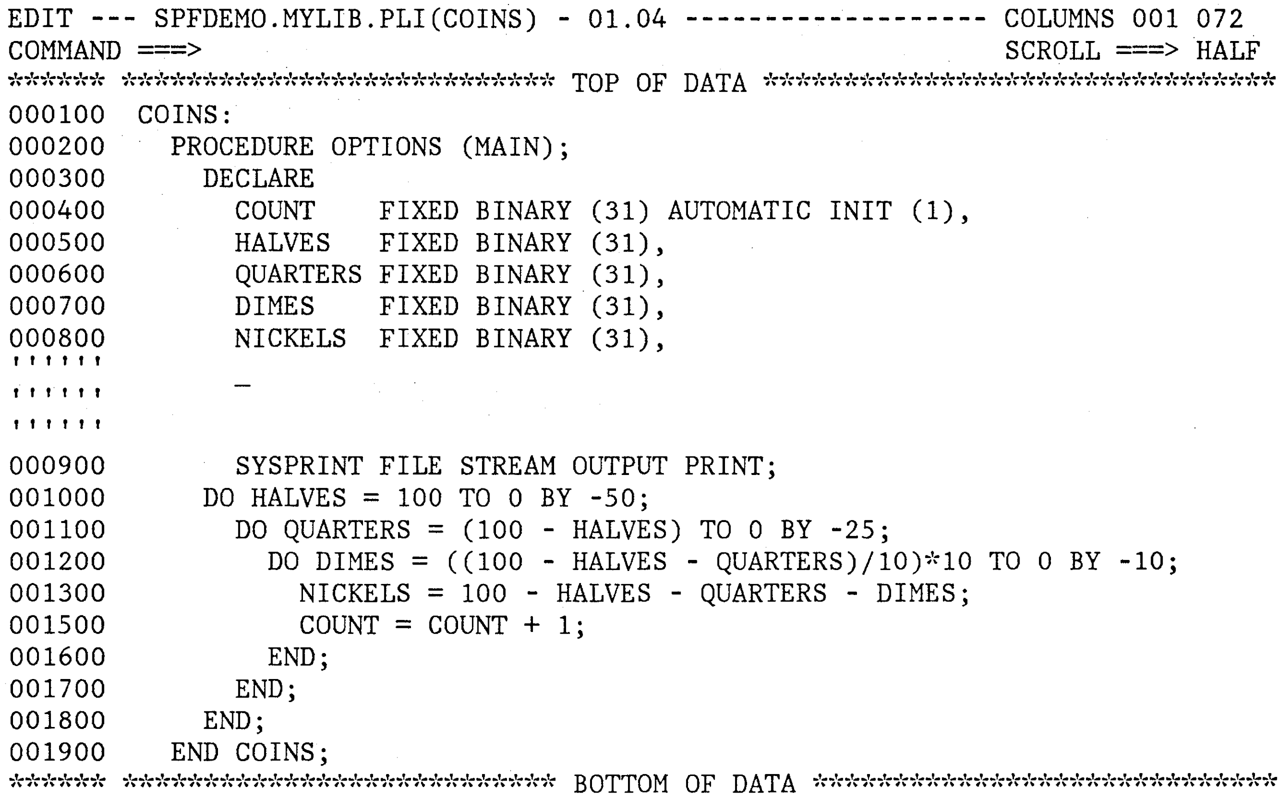
To delete a block of lines, enter the double character "DD" in the line command area of the first and last lines to be deleted. The first and last line need not be on the same page; you can use scrolling between entering the first "DD" and the second "DD".

Figure 32 shows a before-and-after example of line deletion, with line 1400 being deleted.



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------ COLUMNS 001 072
COMMAND ===>                                                 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 ****************************
```

Figure 32 (Part 1 of 2). Edit - Insert and Delete Lines Example



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------- COLUMNS 001 072
COMMAND ===>                                                  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 32 (Part 2 of 2). Edit - Insert and Delete Lines Example

R (Repeat) Command

To repeat a line, enter the single character "R" in the line command area, overtyping the line number. When you press ENTER, the editor inserts a duplicate copy of the line 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. The command has the following format:

```
R    - Repeat a line
Rn   - Repeat a line "n" times
```

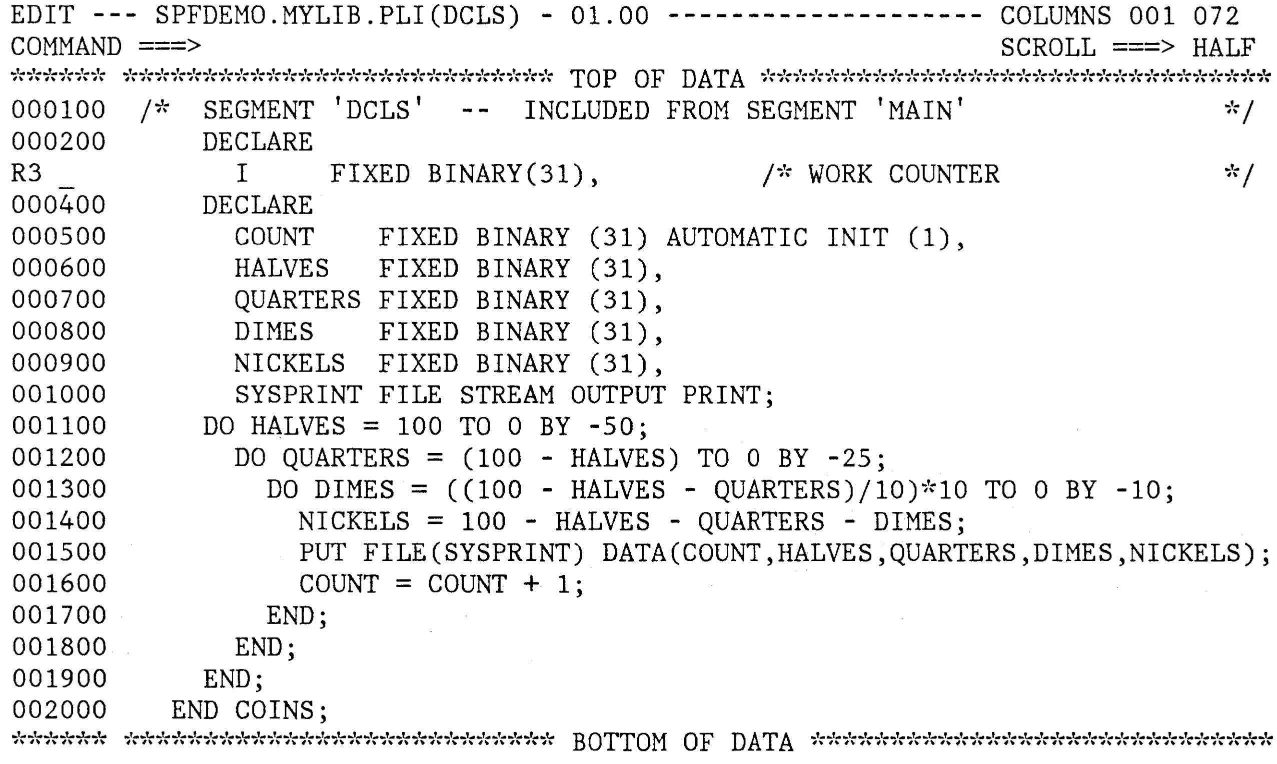
To repeat a block of lines, enter the double character "RR" in the line command area of the first and last lines to be repeated. The first and last lines need not be on the same page. A number may follow the "RR" on either the first or last lines (or both) to indicate that the block of lines is to be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

```
RR   - Repeat a block of lines
RRn  - Repeat a block of lines "n" times
```

As an example of the use of the R and RR commands, assume that the following code is to be generated:

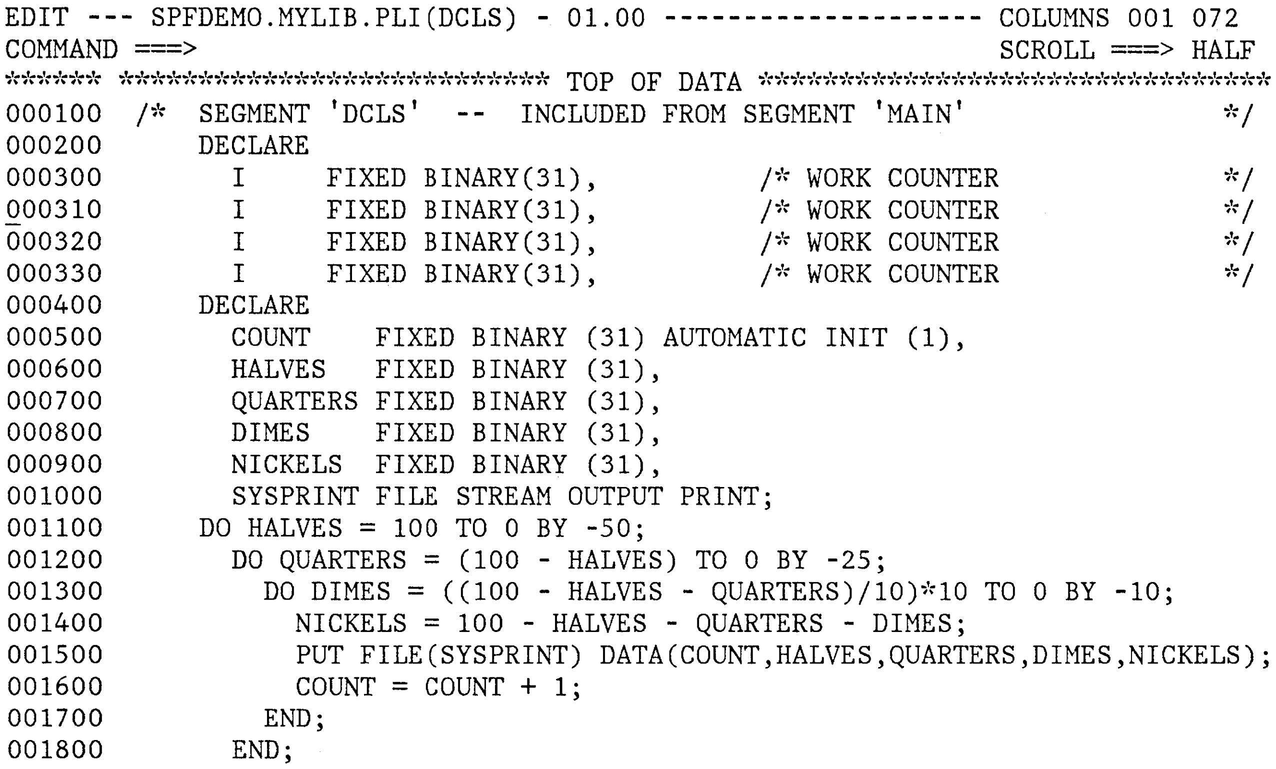
```
DECLARE                                                line 1
   I   FIXED BINARY(31),       /* WORK COUNTER */      line 2
   J   FIXED BINARY(31),       /* WORK COUNTER */      line 3
   K   FIXED BINARY(31),       /* WORK COUNTER */      line 4
   L   FIXED BINARY(31);       /* WORK COUNTER */      line 5
```

The easiest way to generate this code is to type lines 1 and 2, repeat line 2 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 in context is shown in Figure 33.



```
EDIT --- SPFDEMO.MYLIB.PLI(DCLS) - 01.00 ------------------- COLUMNS 001 072
COMMAND ===>                                                  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 ******************************
```

Figure 33 (Part 1 of 2). Edit - Repeat Line Example



```
EDIT --- SPFDEMO.MYLIB.PLI(DCLS) - 01.00 ------------------- COLUMNS 001 072
COMMAND ===>                                                 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;
```

Figure 33 (Part 2 of 2). Edit - Repeat Line Example

M (Move), C (Copy), A (After), and B (Before) Commands

The M and C commands are used to change the location of lines, or to duplicate lines, respectively. The A and B commands are used as pointers to indicate where the moved or copied data is to be placed.

To move a line, enter a single "M" in the line command area of the line to be moved, and enter an "A" (after) in the line command area of another line, indicating the destination of the move. When you press ENTER, the line with the "M" is placed immediately after the line with the "A". Or, you can indicate the destination with a B (before) line command, in which case the line with the "M" is inserted immediately before the line with the "B".

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 panels. The command has the following format:

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

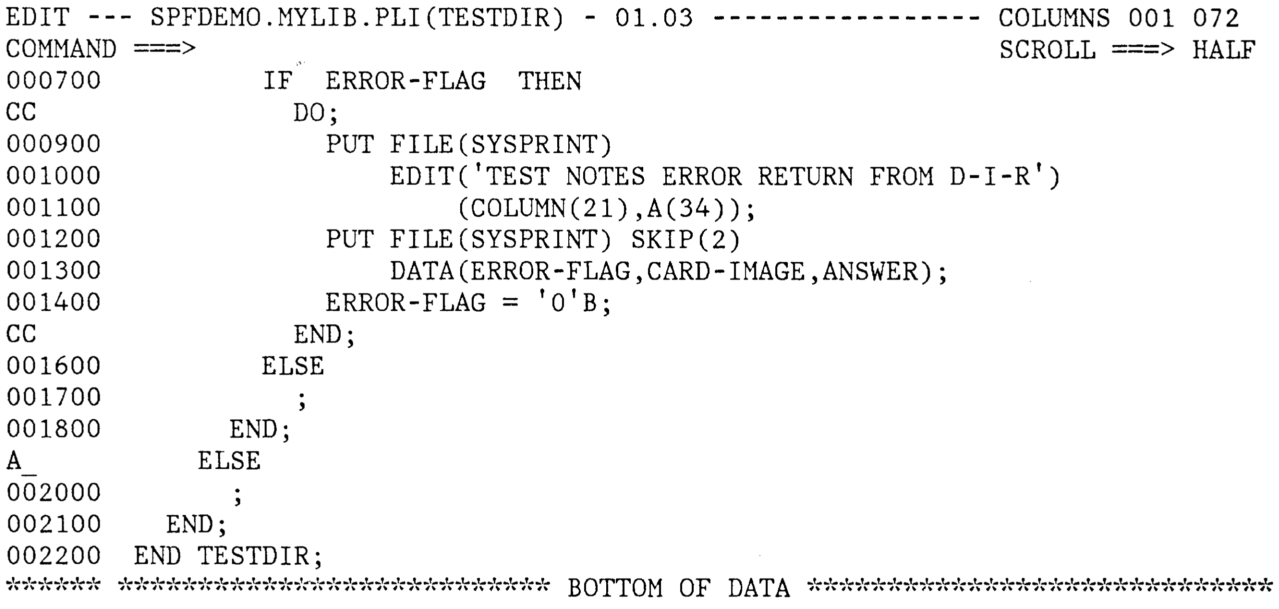
A number may also follow the A or B line command to cause the moved line(s) to be repeated the designated number of times at the point of destination. The commands have the following format:

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

The procedure for copying lines is the same as for moving lines, except that you use a C or CC line command instead of the M or MM. The copy operation leaves the original line(s) in place, and makes a duplicate at the indicated destination. This command has the following format:

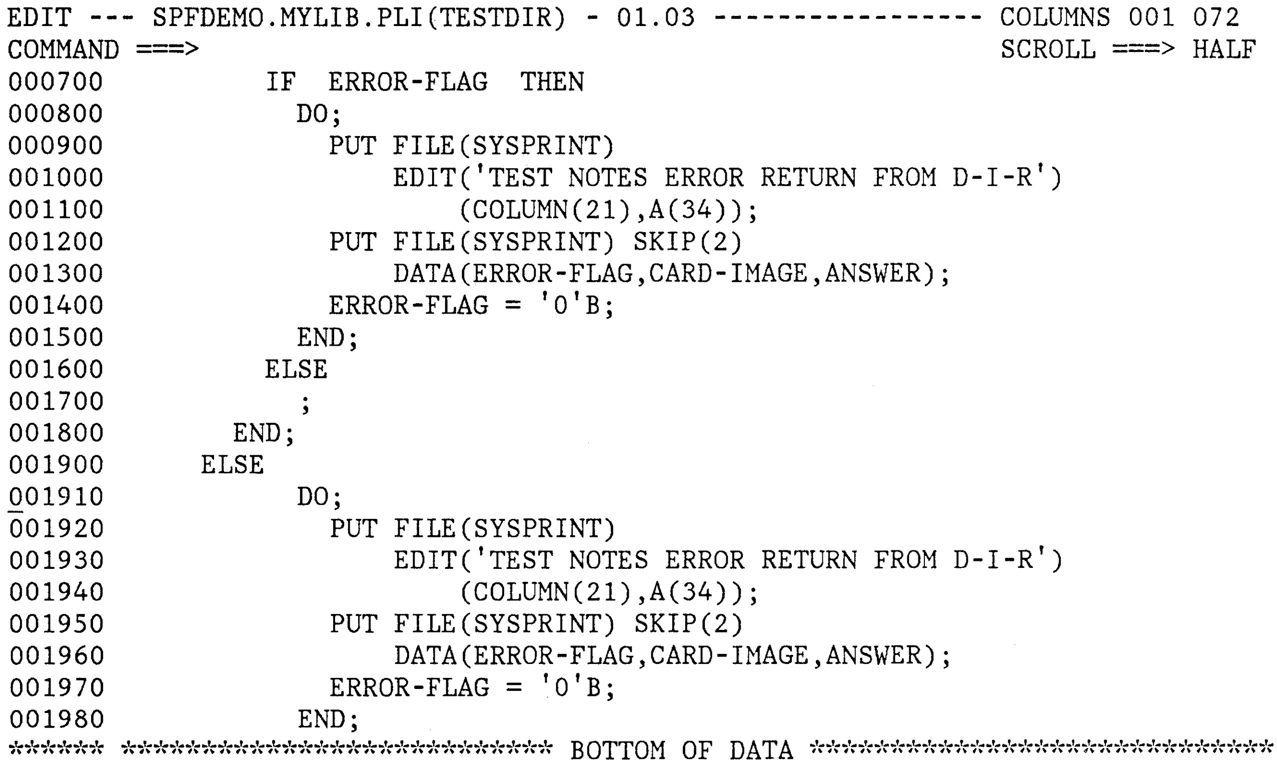
```
C   - Copy a line
Cn  - Copy "n" lines
CC  - Copy a block of lines
```

Figure 34 shows a before-and-after example of copying lines. The block of lines from line 800 to line 1500 is copied after line 1900.



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                 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 ******************************
```

Figure 34 (Part 1 of 2). Edit - Copy Lines Example



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                 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;
****** **************************** BOTTOM OF DATA *****************************
```

Figure 34 (Part 2 of 2). Edit - Copy Lines Example

Preparing Text

You can use all of the edit primary and line commands when entering and editing text information. Three line commands are provided specifically for entering and formatting text:

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

Whenever text line commands are frequently used, you should assign both the TS and TF commands to PF keys. For users of 12-key terminals, infrequently used PF keys may be reassigned to these functions. Use the Parms option to reassign the keys; for example:

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

With these key definitions, you can perform a text split by moving the cursor to the desired split point (within a line) and pressing PF10. After the new material has been typed, pressing PF11 restructures the text from the line containing the cursor to the end of the paragraph.

TE (Text Entry) Command

The TE (text entry) command is intended for bulk entry of text using power typing techniques. When you enter the characters "TE" in the line command area, blank lines are inserted starting with the line following the "TE". Thus, the TE command allows you to enter text without entering the line command I for each text line entered. You may follow the "TE" with a number to indicate the number of blank lines to be inserted. If you omit the number, sufficient blank lines are inserted to fill the screen from the line following the "TE" to the end of the screen. The command has the following format:

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

You can 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 you press ENTER, the new information is automatically reformatted to fit within the currently defined boundaries (see the BOUNDS line command), and any unused blank lines at the end of the new text are deleted. Any words that have been split at line boundaries are automatically put back together when the ENTER key is pressed.

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

> **Note:** If you enter "TEn", where "n" is greater than the number of lines remaining on the screen, the vertical bar is not displayed and the keyboard does not lock at the last character position on the screen. You can scroll down to bring the additional blank text entry lines into view.

You can enter multiple paragraphs in either of two ways:

- Leave a blank line between paragraphs
- Indent the start of a new paragraph with three or more blank (space) characters

> **Note:** Tabs are not functional during text entry.

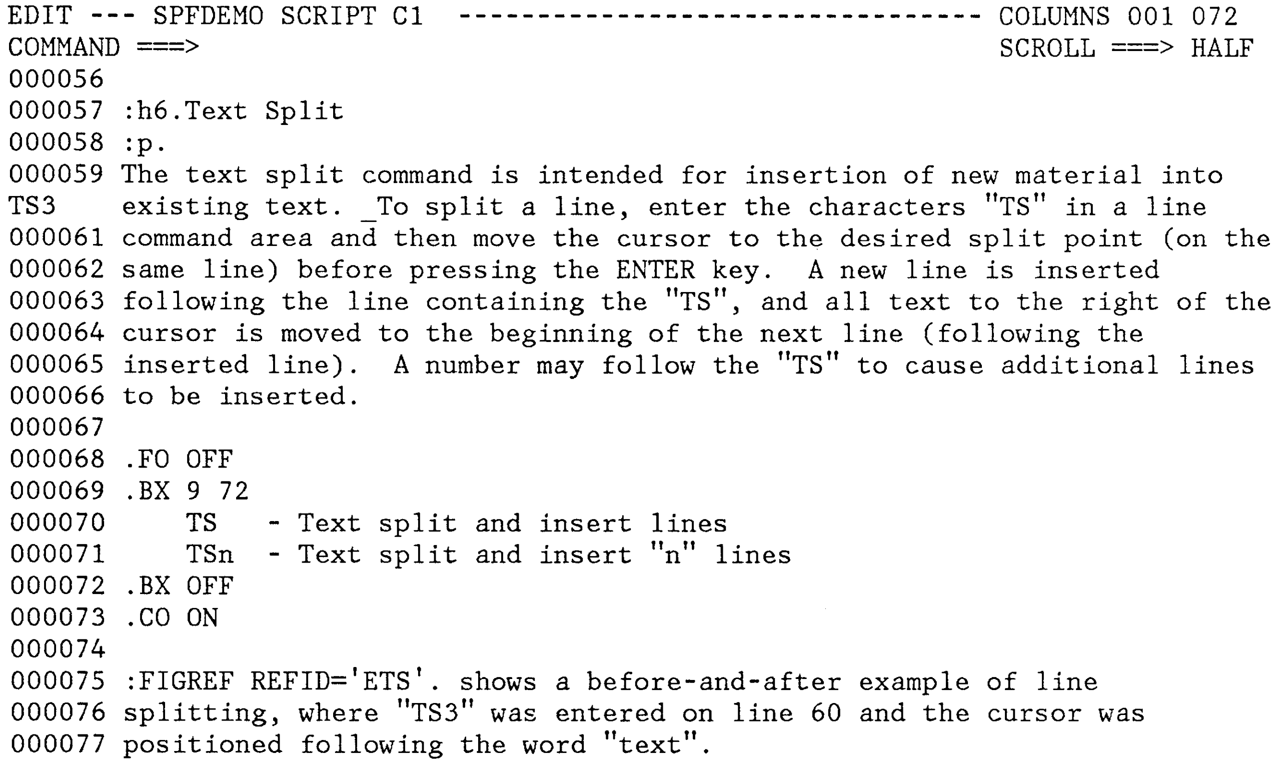
TS (Text Split) Command

The TS (text split) command allows you to add new material to existing text. To split a line, enter the characters "TS" in a line command area, move the cursor to the desired split point (on the same line), and press ENTER. A new line is inserted after the line with the "TS". Text to the right of the cursor, up to and including the current right boundary column, is moved to the next line (after the inserted line) and positioned at the current left boundary column. See the BOUNDS line command for the definition of boundary columns.

You can follow the TS command with a number to cause additional lines to be inserted. The command has the following format:

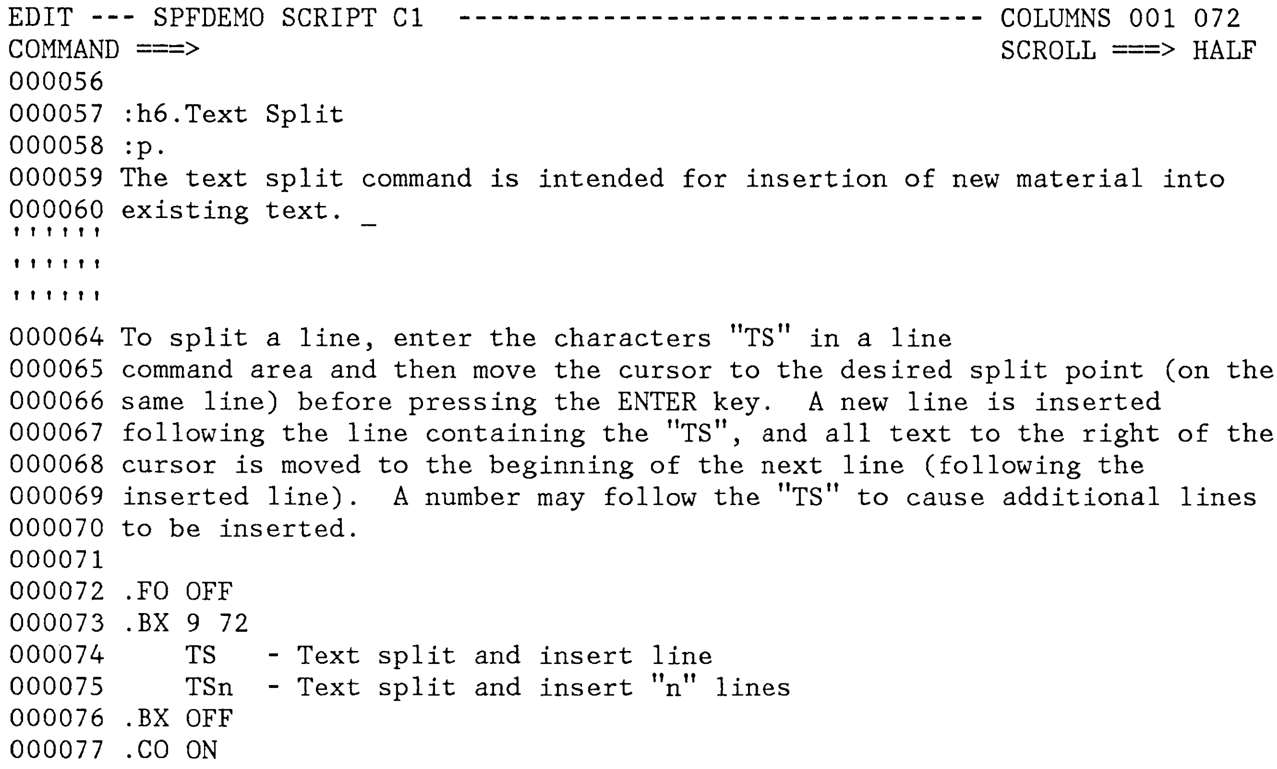
```
TS   - Text split and insert line
TSn  - Text split and insert "n" lines
```

Figure 35 shows a before-and-after example of line splitting. "TS3" was entered on line 60 and the cursor was placed after the word "text".



```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> HALF
000056
000057 :h6.Text Split
000058 :p.
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 lines
000071     TSn  - Text split and insert "n" lines
000072 .BX OFF
000073 .CO 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".
```

Figure 35 (Part 1 of 2). Edit - Text Split Example



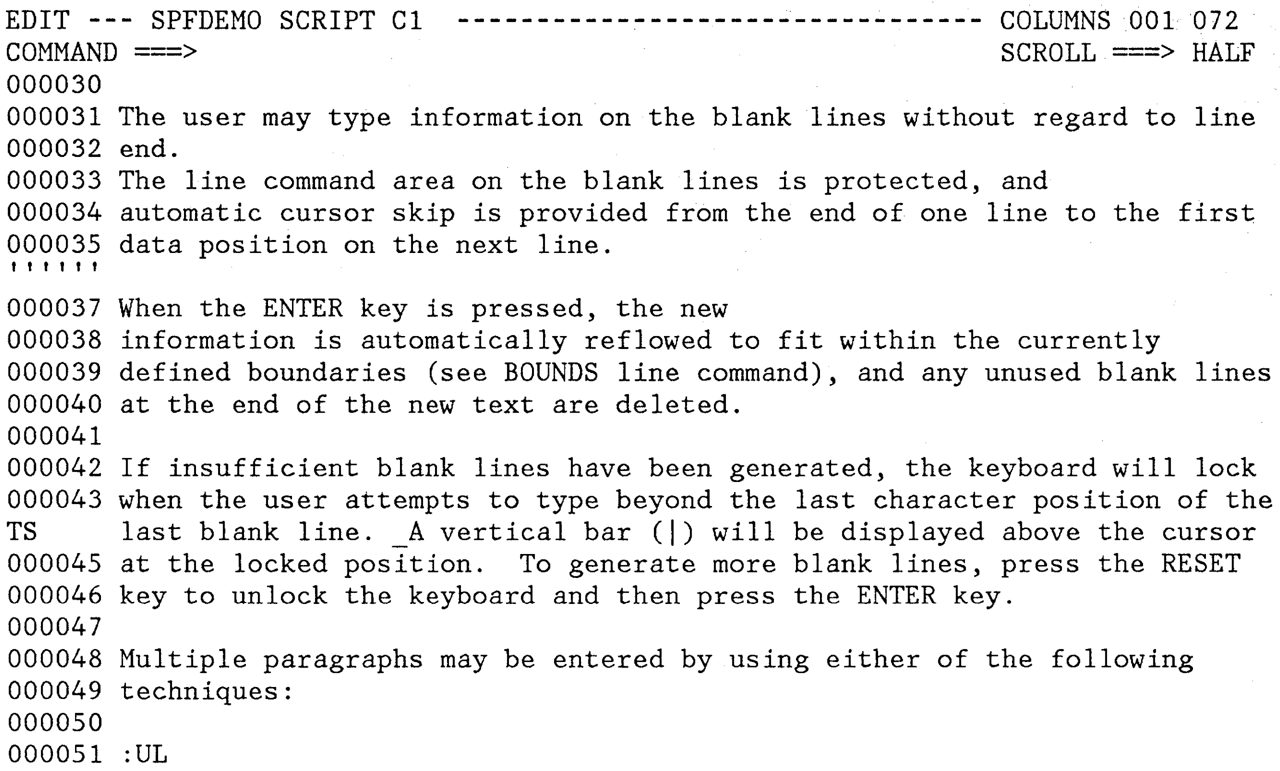
```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> HALF
000056
000057 :h6.Text Split
000058 :p.
000059 The text split command is intended for insertion of new material into
000060 existing text. _
''''''
''''''
''''''
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 .CO ON
```

Figure 35 (Part 2 of 2). Edit - Text Split Example

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

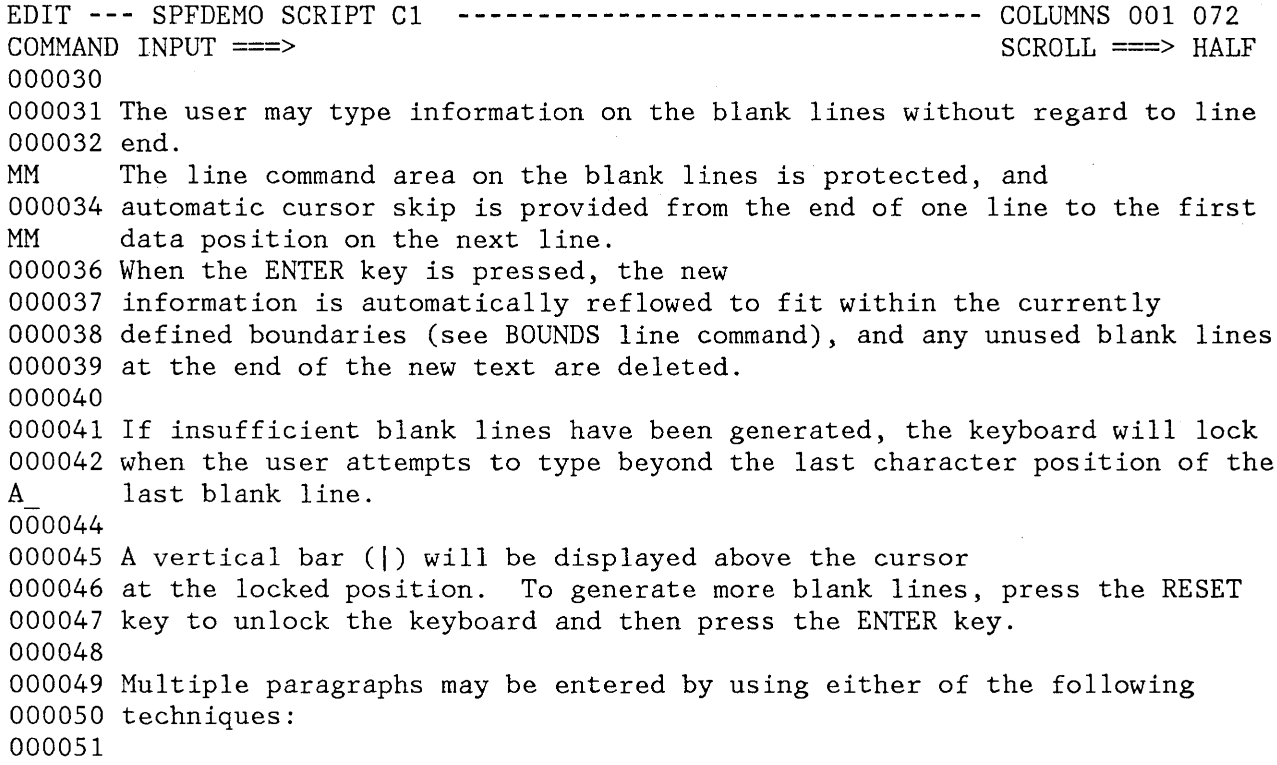
You can also use the TS command with other line commands to help you move or copy text. Enter "TS" at the end points of the material to be moved or copied (i.e., to break it out so that is is wholly contained on one or more lines). In addition, enter "TS" at the insertion point to open up space. Then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

Figure 36 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 isolated 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 commands to accomplish the move.



```
EDIT --- SPFDEMO SCRIPT C1 --------------------------------- COLUMNS 001 072
COMMAND ===>                                                 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
```

Figure 36 (Part 1 of 2). Edit - Move Sentence Example



```
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
```

Figure 36 (Part 2 of 2). Edit - Move Sentence Example

TF (Text Flow) Command

You can use the TF (text flow) command to restructure paragraphs following deletions, insertions, splitting, etc. When you enter the characters "TF" in a line command area, the text is reflowed from the beginning of that line to the end of the paragraph. The end of the paragraph is denoted by a blank line, a change in indentation, or the special characters period (.), colon (:), or ampersand(&). These special characters are used as Document Composition Facility (SCRIPT/VS) control word delimiters. The command has the following format:

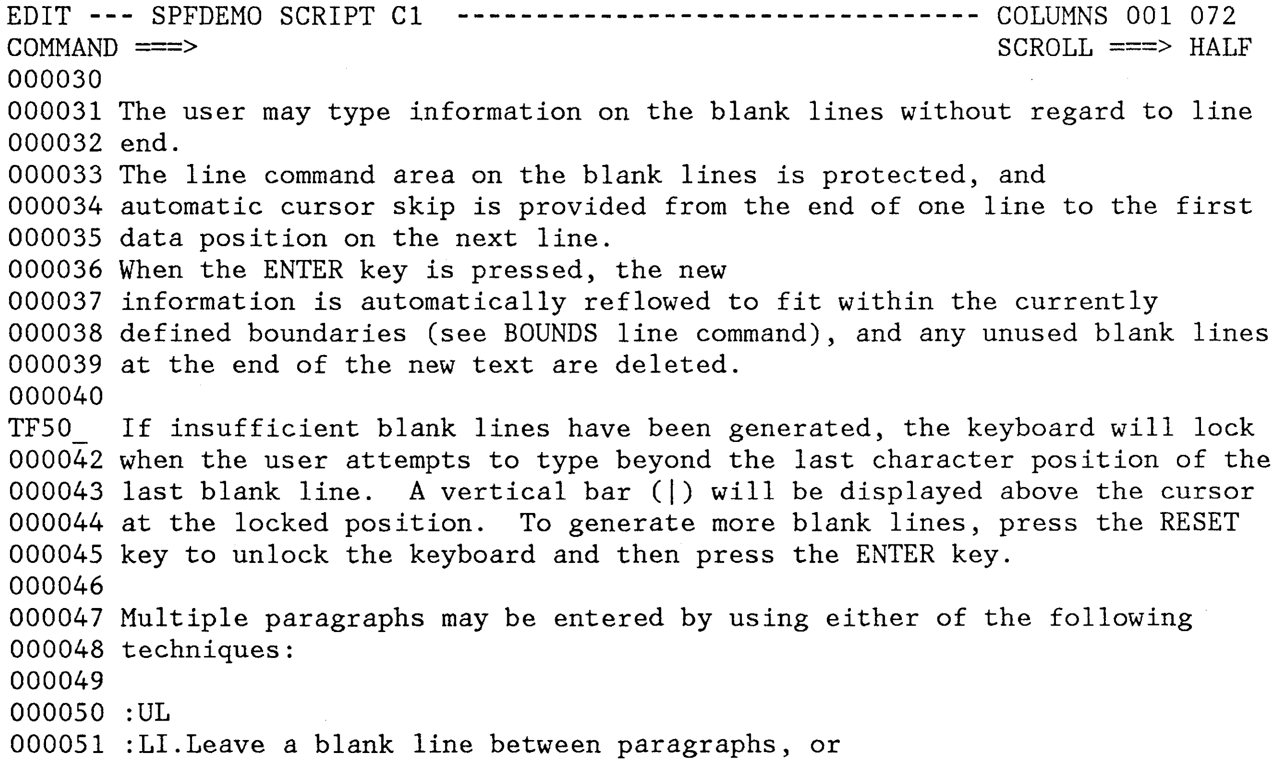
```
TF   - Text flow (between current bounds)
TFn  - Text flow from left bound to column "n"
```

The restructure operation removes trailing blanks on a line by using words from the following line. It does not, however, remove embedded blanks within a line. Accordingly, if one or more words in a line are to be deleted, use the 3270 DELETE key (rather than overtyping the words with blanks).

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

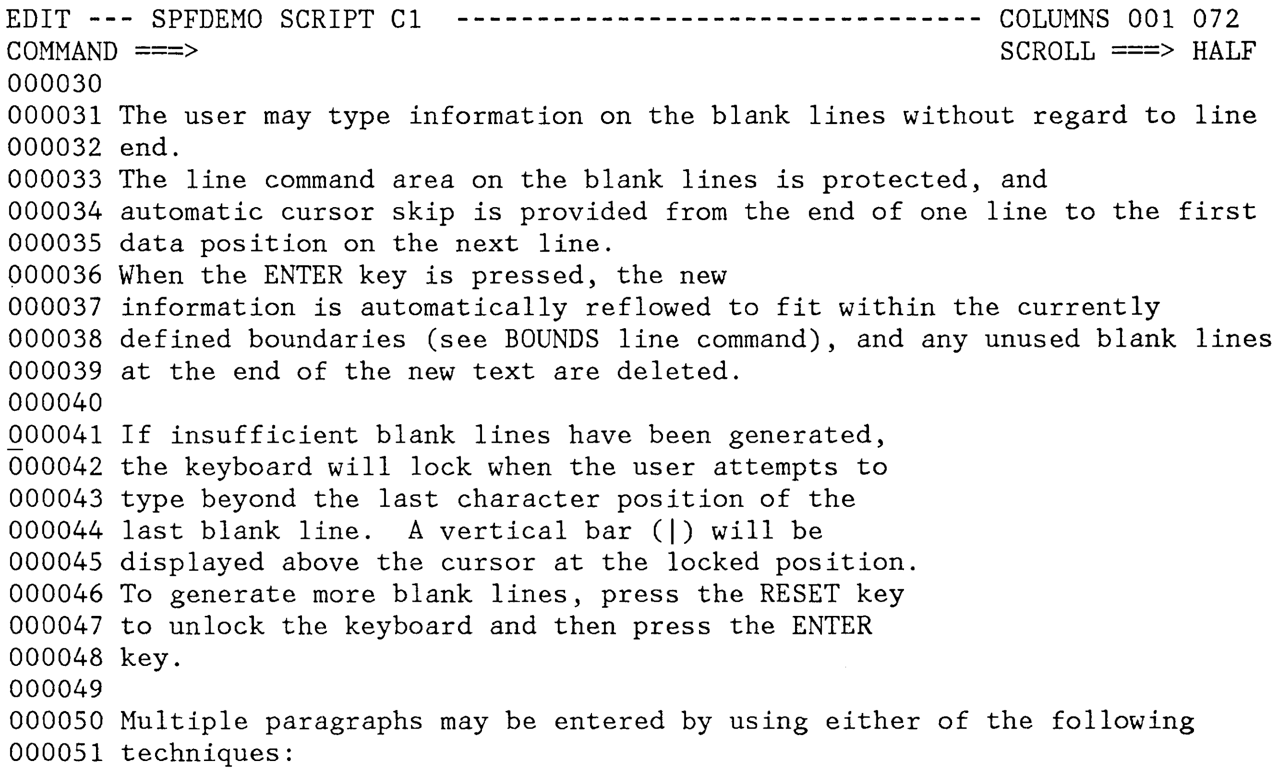
Each line of the restructured text normally extends to the right boundary position. However, you can "squeeze" the text by entering a number after the TF command. The number specifies the rightmost column position for the restructured text. It must be less than the current setting of the right boundary; otherwise, it is ignored.

A before-and-after example of text restructuring is shown in Figure 37. The bounds are set at columns 1 and 72. A "TF50" command is entered on line 41. All text between columns 1 and 72 participates in the flow, but the results are contained within columns 1 and 50.



```
EDIT --- SPFDEMO SCRIPT C1  -------------------------------- COLUMNS 001 072
COMMAND ===>                                                 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
```

Figure 37 (Part 1 of 2). Edit - Text Restructure Example



```
EDIT --- SPFDEMO SCRIPT C1 -------------------------------- COLUMNS 001 072
COMMAND ===>                                                 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:
```

Figure 37 (Part 2 of 2). Edit - Text Restructure Example

Overlaying Lines (O Command)

The O command allows you to rearrange a single-column list of items into multicolumn (tabular) format. You can use the O command with the M or C commands instead of the A command.

Enter the character "O" in the line command area to specify the line over which data is to be moved or copied. Enter a number following the "O" to indicate the number of lines to be overlayed. To overlay a block of lines, enter "OO" on the first and last lines of the block. The command has the following format:

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

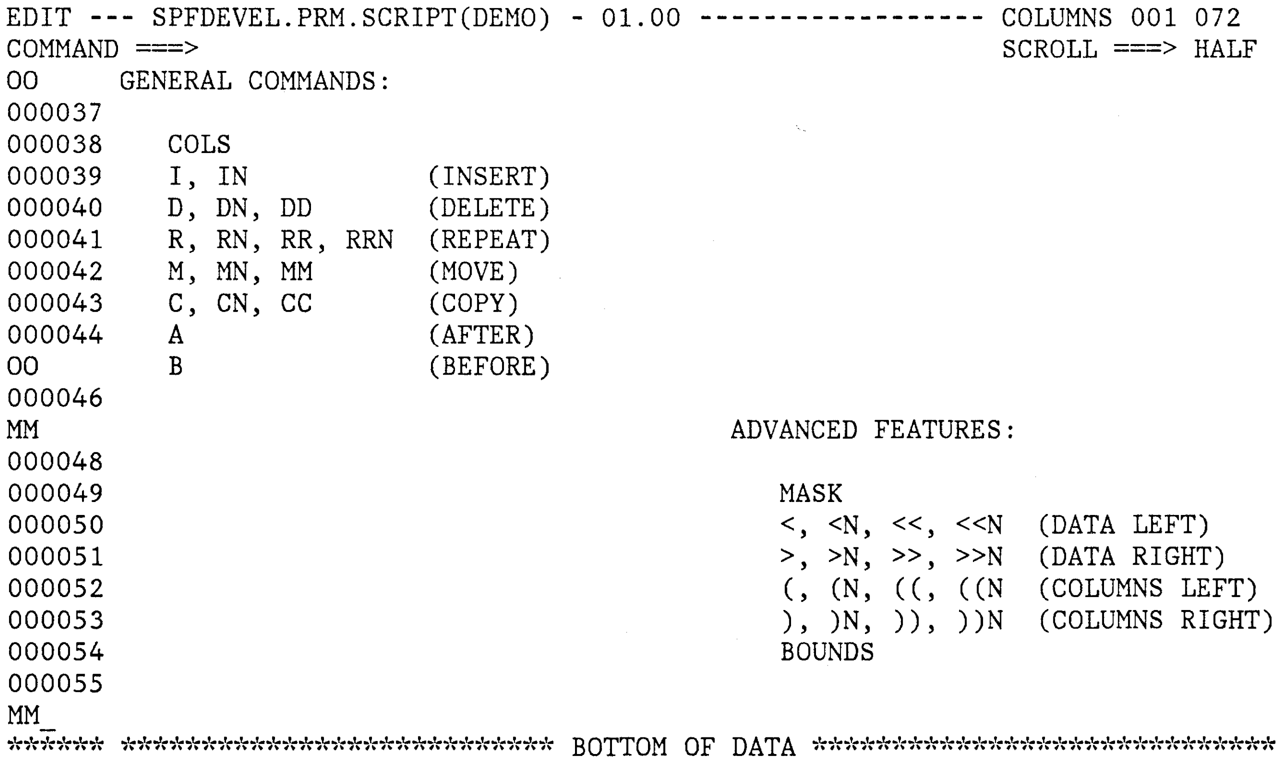
Only blank characters in the receiving line(s), specified with "O" or "OO", are overlaid with corresponding characters from the source line(s), as specified by the M, MM, C, or CC commands. Nonblank characters are not overlaid.

Only those characters that are within the current column boundaries participate in the overlay operation (see the description of the 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.

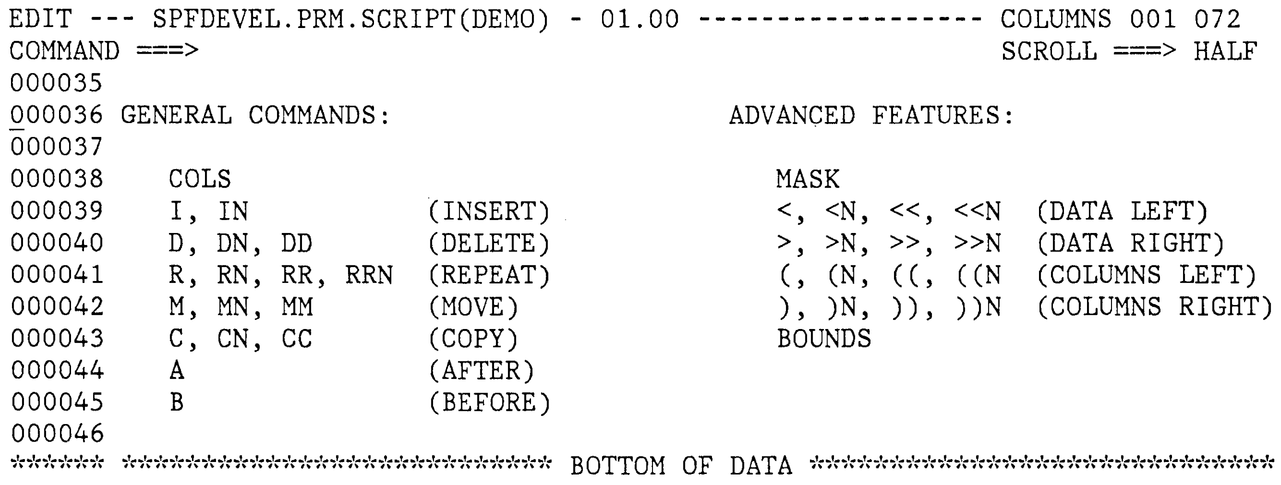
Only data lines participate in the overlay operation. Special lines such as MASK, TABS, BOUNDS, and COLS are ignored as either source or receiving lines.

A before-and-after example of overlaying lines is shown in Figure 38. 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 multicolumn format.



```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.00 ----------------- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> HALF
00     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)
00        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 *****************************
```

Figure 38 (Part 1 of 2). Edit - Overlay Line Example



```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.00 ----------------- COLUMNS 001 072
COMMAND ===>                                                  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 38 (Part 2 of 2). Edit - Overlay Line Example

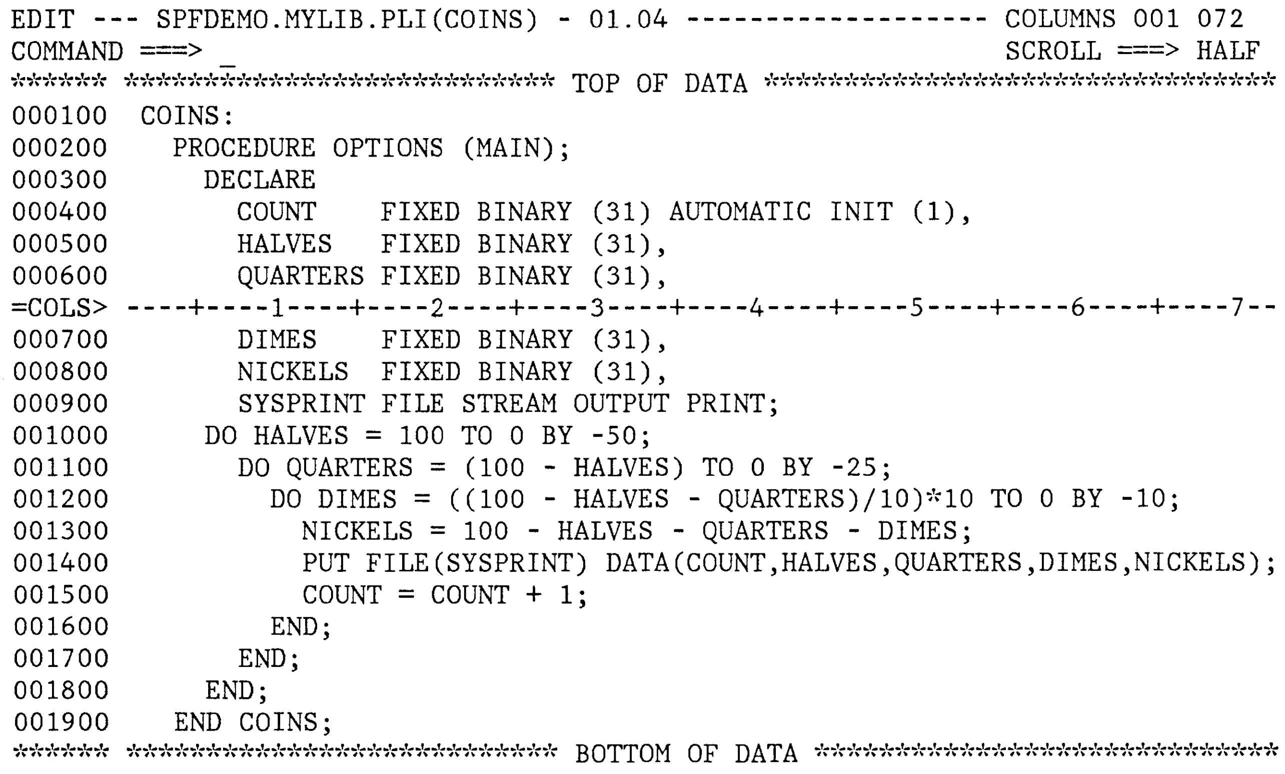
Identifying Columns (COLS Command)

You can display a position identification line by entering the COLS command in the line command area. The command has the following format:

```
COLS
COL
```

When you press the ENTER key, a special line is inserted at the designated position and the remaining lines are pushed down one. An example of the COLS line is shown in Figure 39. The characters '=COLS>' in the line command area are displayed by edit to identify the COLS line.

You can remove the column identification line from the 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.



```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ------------------ COLUMNS 001 072
COMMAND ===> _                                              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 39. Edit - Column Identification Line

Defining Masks (MASK Command)

You can use a mask 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. You can define the mask by entering characters when the blank mask is displayed.

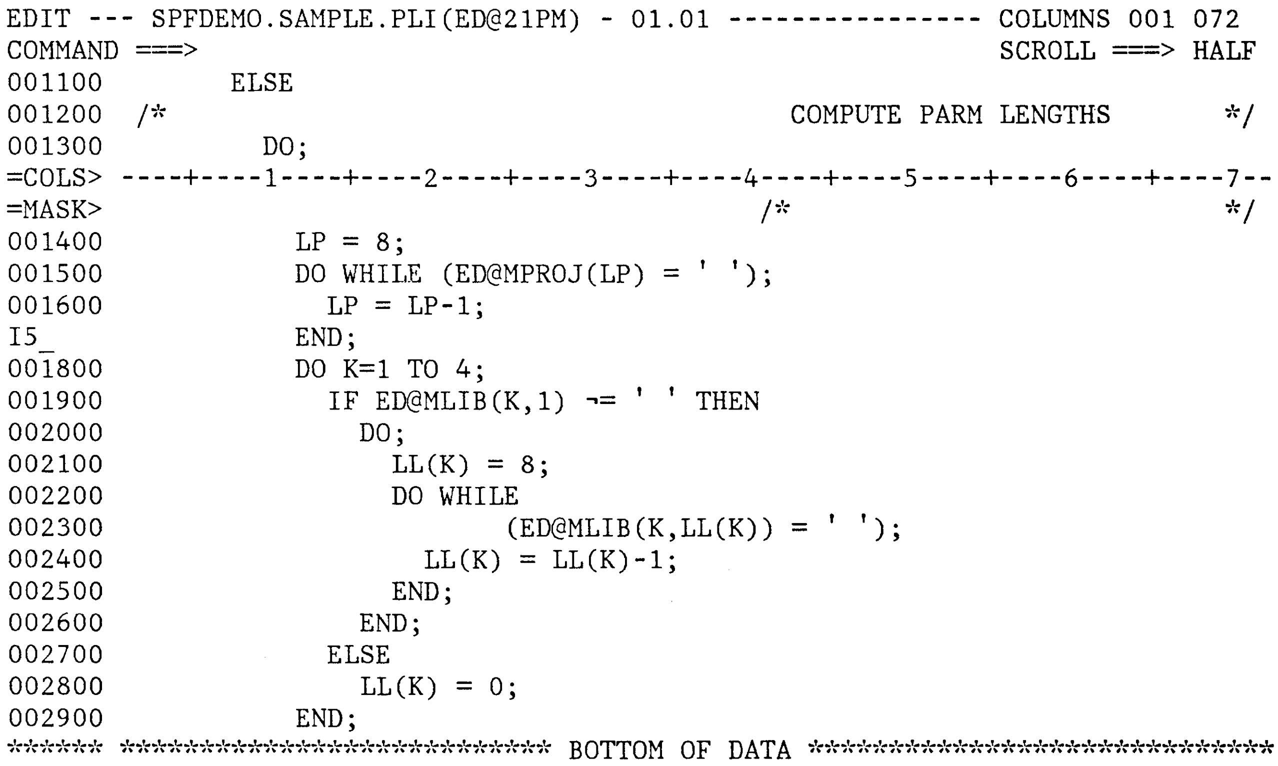
To display the mask, enter the characters "MASK" in the line command area, overtyping the line number. When you press ENTER, the mask is displayed at the designated position.

MASK - Display mask definition line

Once you have displayed the mask, you can change it to allow pre-entry of data on inserted lines. An example is shown in Figure 40. 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.

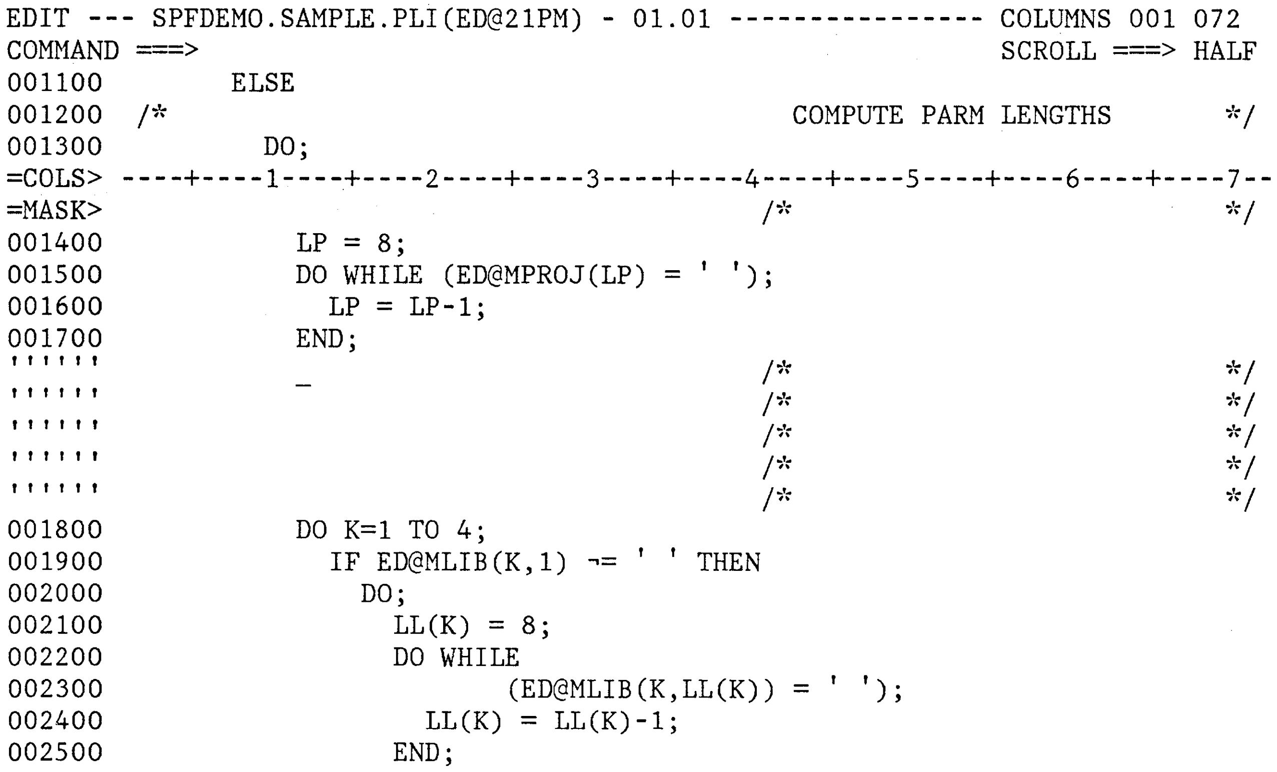
To remove the mask line from the display, enter a "D" in the line command area, or enter a RESET primary command. The mask line is never saved as part of the data.

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



```
EDIT --- SPFDEMO.SAMPLE.PLI(ED@21PM) - 01.01 ---------------- COLUMNS 001 072
COMMAND ===>                                                  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;
****** ****************************** BOTTOM OF DATA ******************************
```

Figure 40 (Part 1 of 2). Edit - Define Mask Example



```
EDIT --- SPFDEMO.SAMPLE.PLI(ED@21PM) - 01.01 ---------------- COLUMNS 001 072
COMMAND ===>                                                  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;
```

Figure 40 (Part 2 of 2). Edit - Define Mask Example

Controlling Tabs (TABS Line Command)

Use the TABS line command to define tab positions for software, hardware, and logical tabs. To display the tab definition line, enter the characters "TABS" or "TAB" in the line command area, overtyping the line numbers. When you press ENTER, the tab definition line is inserted at the designated position.

```
TABS  - Display tab definition line
TAB
```

To define software tab positions, type an underscore (_) or hyphen (-) at each desired column position on the tab definition line. Type two or more consecutive underscores or hyphens to define a software tab field, which causes the cursor to be repositioned to the first nonblank character within the field. If the tab field contains all blanks, the cursor is positioned at the beginning of the field.

To define hardware and logical tab positions, type an asterisk (*) at each desired column position on the tab definition line. The asterisks indicate the locations at which the attribute bytes will be inserted (for hardware tabs) or the locations following which strings will be repositioned (for logical tabs). For example, if your tab character is $ and you enter:

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

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

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

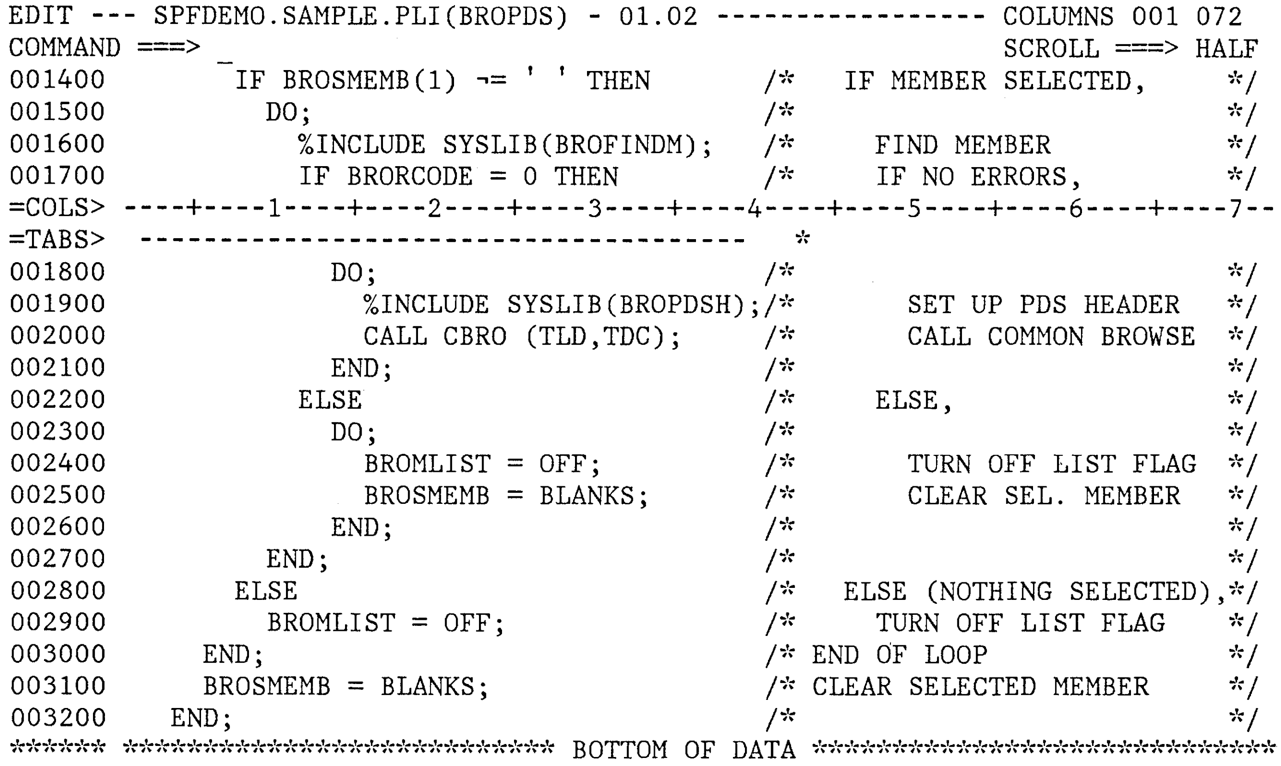
Initially, the tab definition line contains all blanks. If you display the tab line and enter software tab definitions, they take effect immediately; each line contains a software tab or tab field at the designated column positions. Hardware and logical tab definitions do not take effect immediately. 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. You can enter or leave tabs mode using the TABS primary command.

Figure 41 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.

To remove the tab definition line from the display, enter a "D" in the line command area, or enter 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 you change them. Tab definitions are retained in the current edit profile, and are automatically used the next time you edit the same type of data.

Refer to the discussion of the TABS primary command for information on defining the tabs character and setting tabs mode.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND ===> _                                                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 41. Edit - Tab Definition Line

Defining Boundary Columns (BOUNDS Command)

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

```
BOUNDS  - Display boundary definition line
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 are not specified

- TE (text entry), TS (text split), and TF (text flow) line commands

- Overlay line command

- Left and right scrolling

The effect on left and right scrolling is that a left scroll stops at the left bound, and a right scroll stops at the right bound. A subsequent left or right scroll goes beyond the bound (assuming the bound is not at the leftmost or rightmost column position).

The boundaries are initialized by edit based on the data "type" (the ISPF 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 42.

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

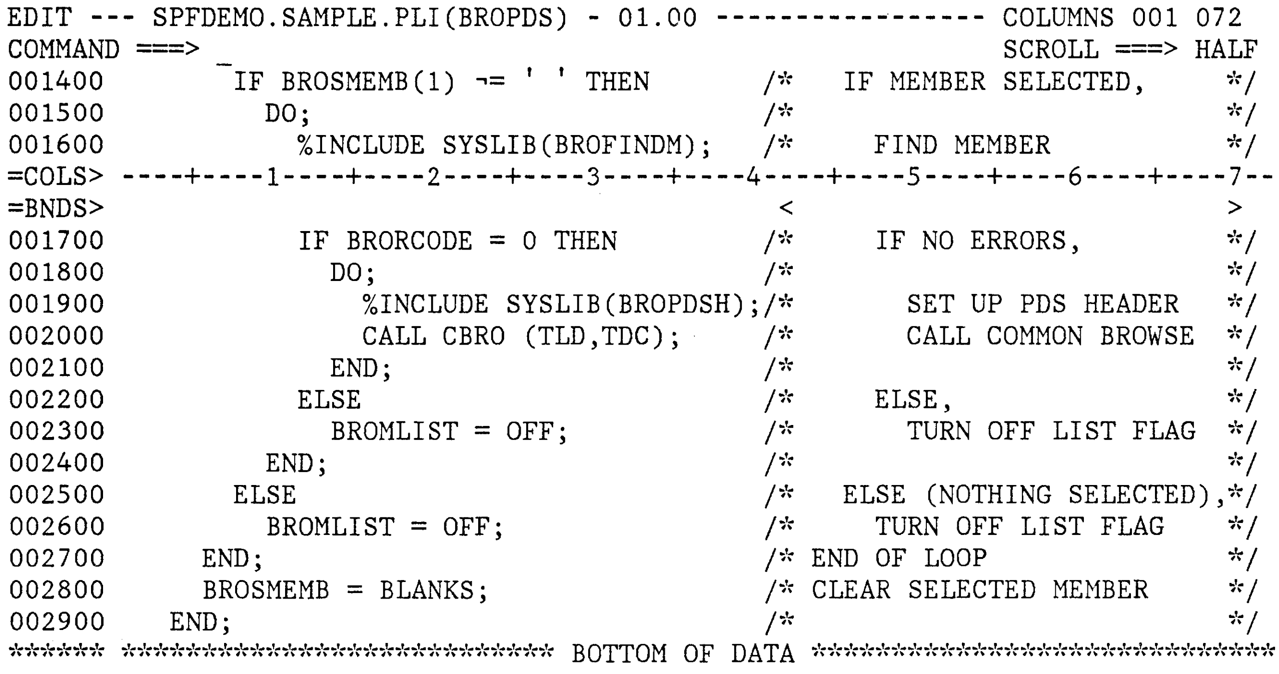
Figure 42. Edit - Default Column Boundary Settings

You can 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 you have already changed the boundaries, you can revert to the default bounds by displaying the boundary definition line and blanking out its contents with the ERASE EOF key.

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

Figure 43 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.

To remove the boundary definition line from the display, enter a "D" in the line command area, or enter a RESET primary command. The boundary definition line is never saved as part of the data.



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.00 ----------------- COLUMNS 001 072
COMMAND ===> _                                                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 43. Edit - Boundary Definition Line

Changing Indentations

Use the shift line commands to shift the contents of a line (or block of lines) left or right. The 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 prevents loss of data. Column shifting shifts information within column boundaries without regard to its contents, and may result in the loss of data.

To shift the contents of a line left, enter the single character "<" or "(" in the line command area, overtyping the line number. When you press ENTER, 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

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

To shift a block of lines left, enter "<<" or "((" 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; you can scroll 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

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

You can enter a number 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 you can change. Default column boundaries are established by edit, based on the data "type". The default boundaries are normally the first and last columns where source text may be entered for the particular programming language. See the BOUNDS command for a discussion of default boundaries and the procedures for changing them.

> and < (Data Shifting) Commands

Within the column boundaries, data shifting is accomplished by removing multiple blank characters at one end of the data, and adding blanks at the other end. The following rules apply:

- Nonblank characters are never deleted or truncated.

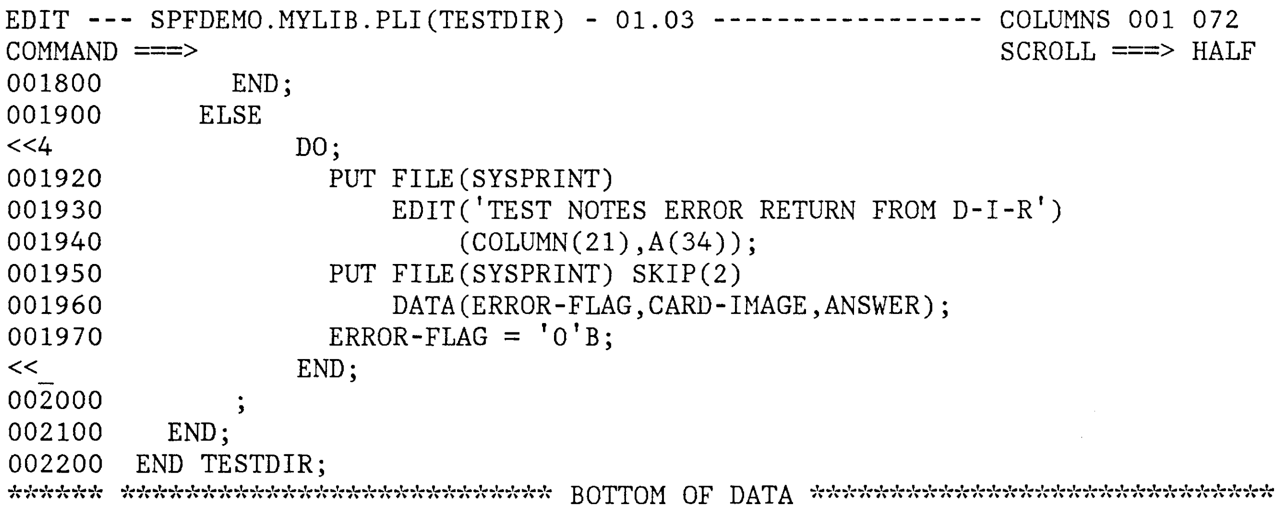
- A single blank is never deleted.

- Within apostrophes, the number of multiple blanks is never changed.

- Nonblank characters are never shifted into or out of the left column boundary position (normally the label field).

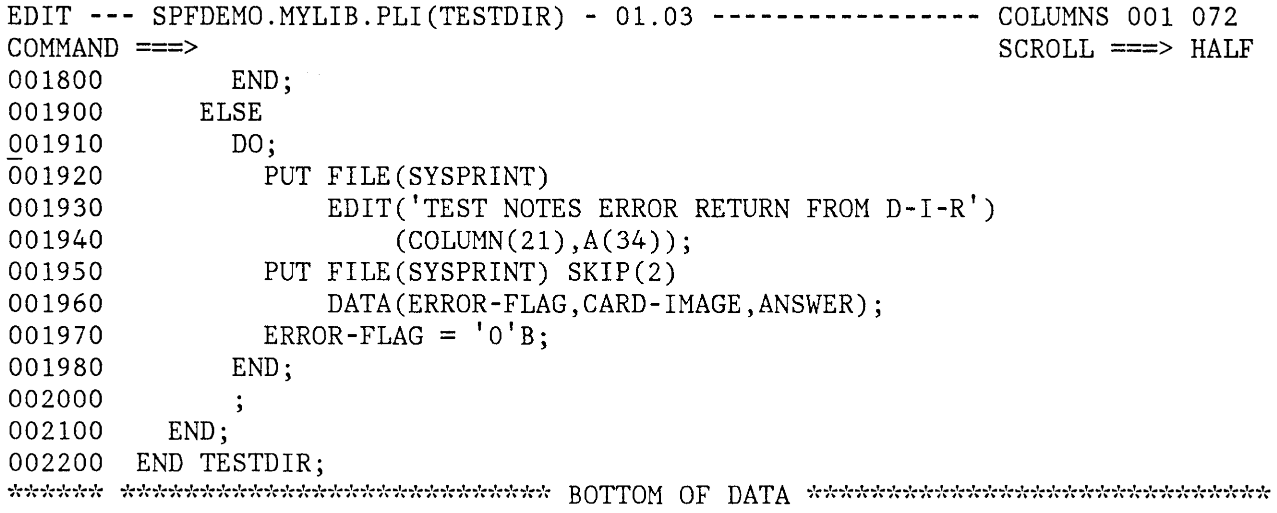
- If a shift cannot be completed, it is partially performed and the line number is replaced by the intensified warning message: ==ERR>. You can remove the message with the RESET primary command, or you can overtype the message or data on that line.

Figure 44 shows a before-and-after example of data shifting. Lines 1910 through 1980 are shifted left four column positions.



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                  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 ******************************
```

Figure 44 (Part 1 of 2). Edit - Left Shift Data Example



```
EDIT --- SPFDEMO.MYLIB.PLI(TESTDIR) - 01.03 ---------------- COLUMNS 001 072
COMMAND ===>                                                 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 44 (Part 2 of 2). Edit - Left Shift Data Example

) and ((Column Shifting) Commands

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 nonblank characters into or out of a column boundary position. Characters shifted beyond a column boundary position are lost, with no warning message.

Excluding Lines (X, F, L, and S Commands)

When editing a program segment that exceeds the display screen size, it is often difficult to determine whether the control structure and indentation levels are correct. The ability to exclude lines allows you to remove blocks of lines from the display so that you can see the overall control structure. The lines are excluded from the display but are not deleted from the data.

You can also use excluded lines with the X and NX options of the FIND and CHANGE commands to limit the scope of search, or to easily identify lines that have been found or changed, by causing them to "pop out" of a block of excluded lines.

To exclude a line, enter the single character "X" 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 you press ENTER, the line(s) are replaced by a single message line, which indicates how many lines were excluded.

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

To exclude a block of lines, enter the characters "XX" 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 45 shows a before-and-after example of excluded lines. The block of lines, starting at line 1600 through line 2600, is excluded from the display and replaced by a single message line.

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

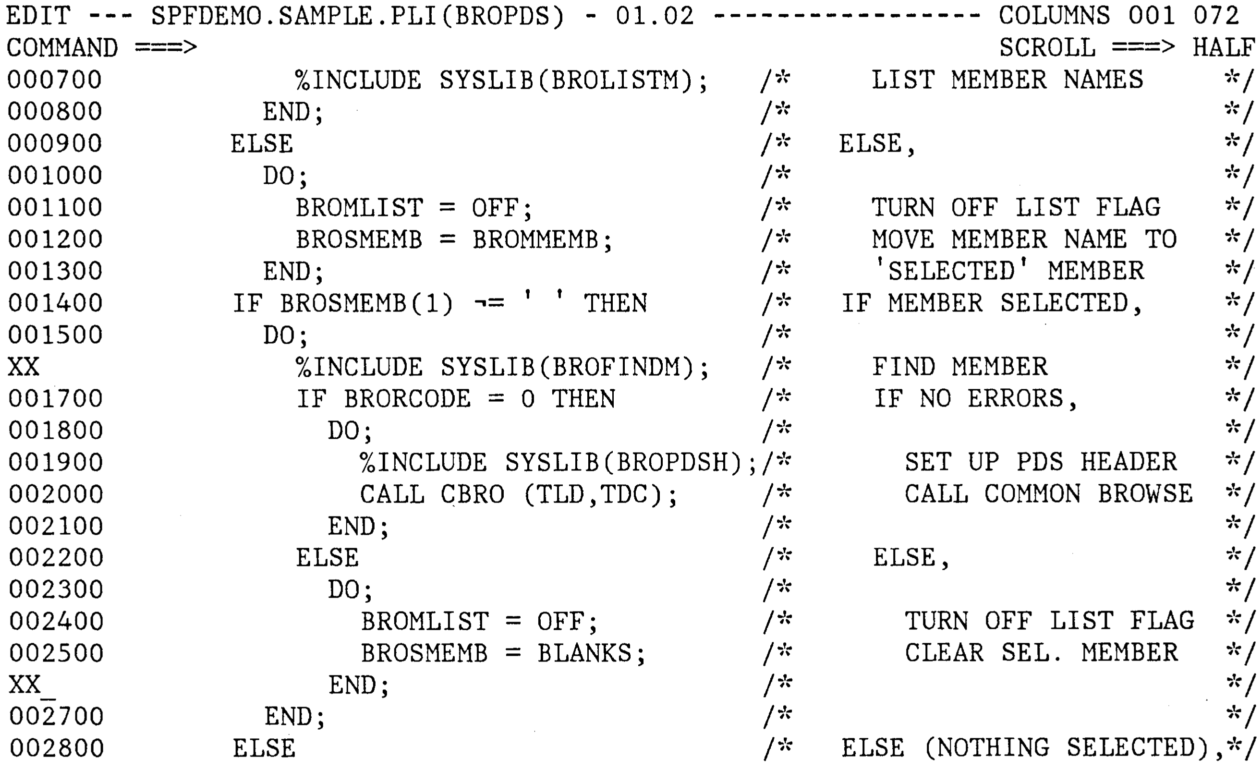
S - Show a line
Sn - Show "n" lines

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

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

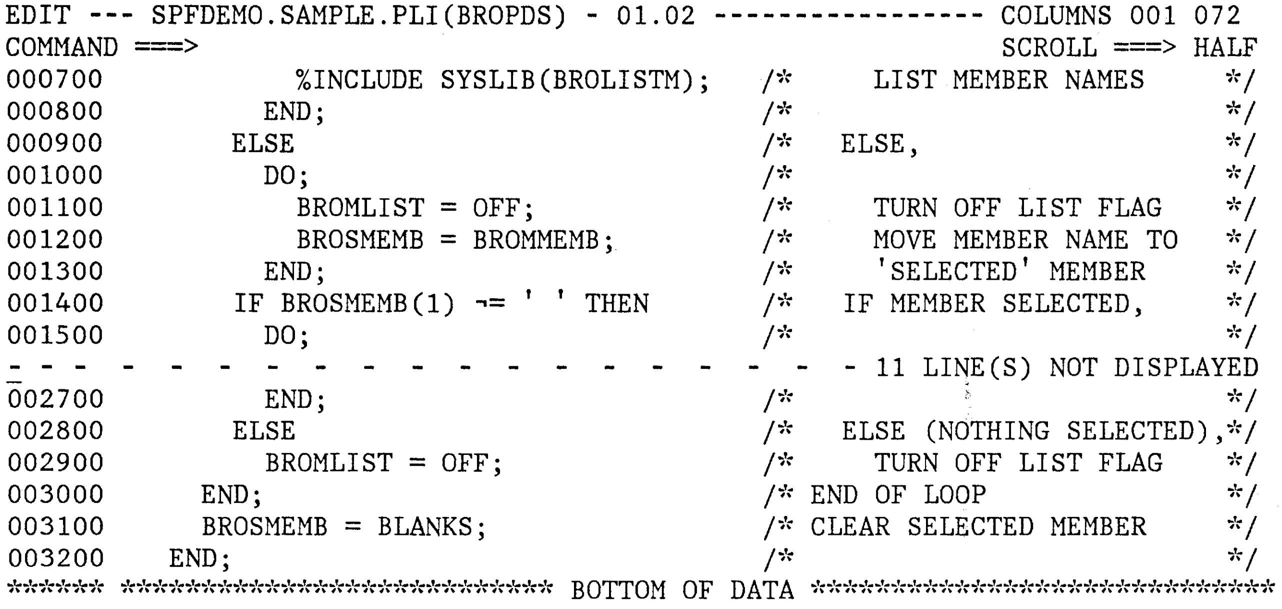
The S line command causes the block of excluded lines to be scanned, and one or more lines selected for redisplay. The selected lines are those with the leftmost indentation levels (i.e, with the fewest number of leading blanks). If "S3" is entered, for example, the three lines with the leftmost indentation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



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND ===>                                                  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),*/
```

Figure 45 (Part 1 of 2). Edit - Excluded Lines Example



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ---------------- COLUMNS 001 072
COMMAND ===>                                                  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 45 (Part 2 of 2). Edit - Excluded Lines Example

Figure 46 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.

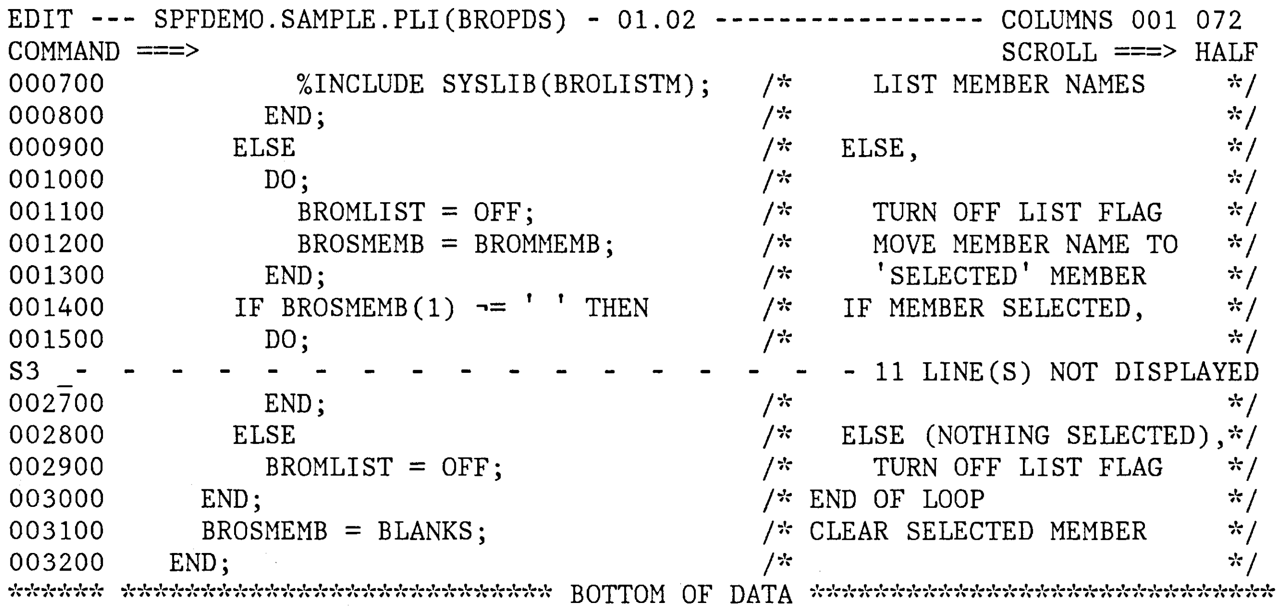
You can enter any line command that normally operates on a single line in the line command area of the "EXCLUDED LINES" message. For example, a "D" entered on that line deletes the entire block of excluded lines. This feature is particularly useful in conjunction with the shift commands. Suppose, for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

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

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

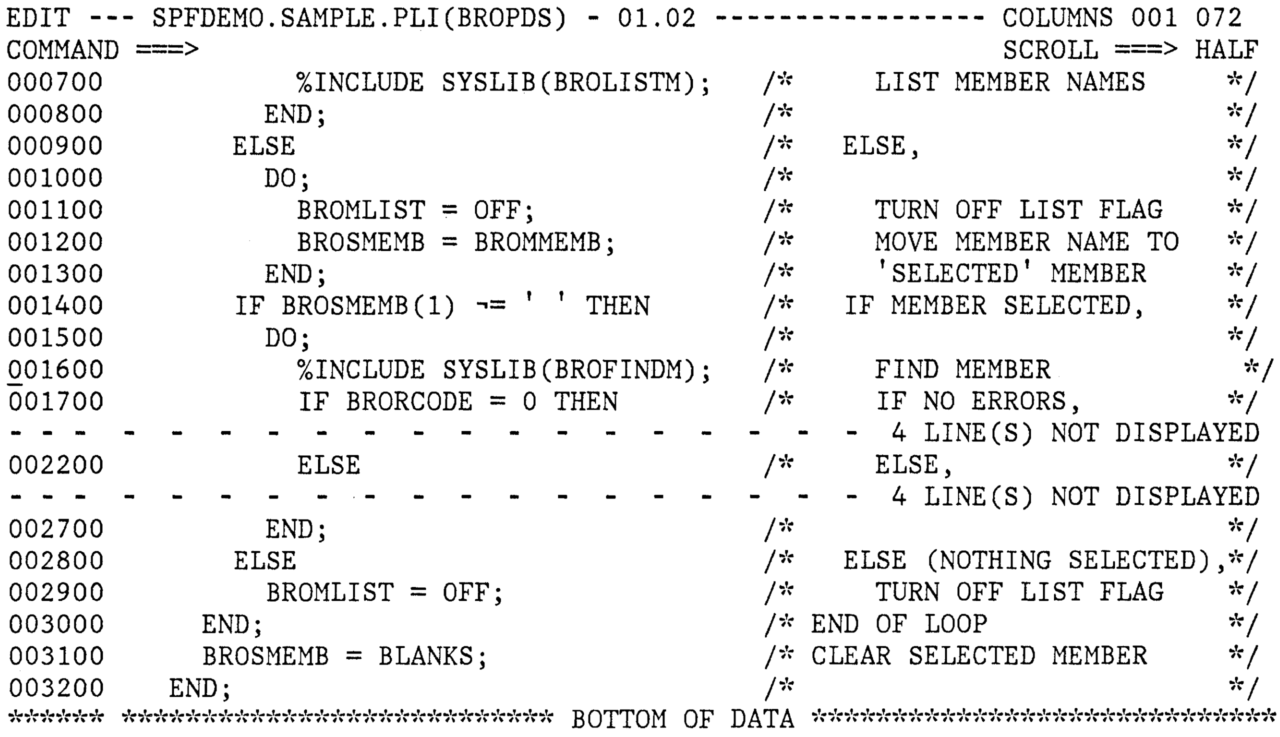
3. In the primary command area, enter the RESET command to bring the lines back into view.

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



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ----------------- COLUMNS 001 072
COMMAND ===>                                                   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 ******************************
```

Figure 46 (Part 1 of 2). Edit - Show Lines Example



```
EDIT --- SPFDEMO.SAMPLE.PLI(BROPDS) - 01.02 ---------------- COLUMNS 001 072
COMMAND ===>                                                 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 46 (Part 2 of 2). Edit - Show Lines Example

DIALOG DEVELOPMENT MODELS

To assist you in writing dialogs, models for various parts of a dialog are provided in the skeleton library, ISPSLIB. A model is a predefined set of statements for a dialog element (for example, a call to an ISPF service) that you can copy into the file you are editing, and then modify for your specific application.

Each model contains three parts:

1. Executable source statements - This is the actual information (dialog service call, panel format, etc.) that will be used in the data set being edited. Fields in the source statements may be updated by inserting names, parameters, etc.

2. Comments - Source statement comments, in the form valid for the type of model, are included for models of dialog service calls to document the meanings of the possible return codes from the service. These comments allow you to develop error handling logic for your

function. They may also provide parameter descriptions for other types of models.

3. Notes - The notes provide tutorial information about what the model does, and what might be entered in each variable field in the source statements. The notes are not stored with the edited data set and so do not appear in the edited data set after it has been saved or filed, or after the edit RESET command has been issued.

When you file or save the data, only the source statements themselves and any comments are saved.

Models are distributed with PDF for panels, messages, skeletons, and command and program invocation of ISPF services.

Models are organized into a hierarchy based on the type and version of the dialog element they represent. This hierarchy is represented by a logical name for each model, where each part of the name corresponds to a level in the hierarchy.

The first part of the logical name is the model **class**. There is a model class for each data set type qualifier in which a dialog element might be stored. The classes that have been defined for the models distributed by IBM are listed on the selection panel shown in Figure 47. You are prompted by this panel when you need to set the desired model class, if you do not name the class explicitly.

You can default this part of the logical name whenever the type qualifier of the file you are editing matches the class of the model desired.

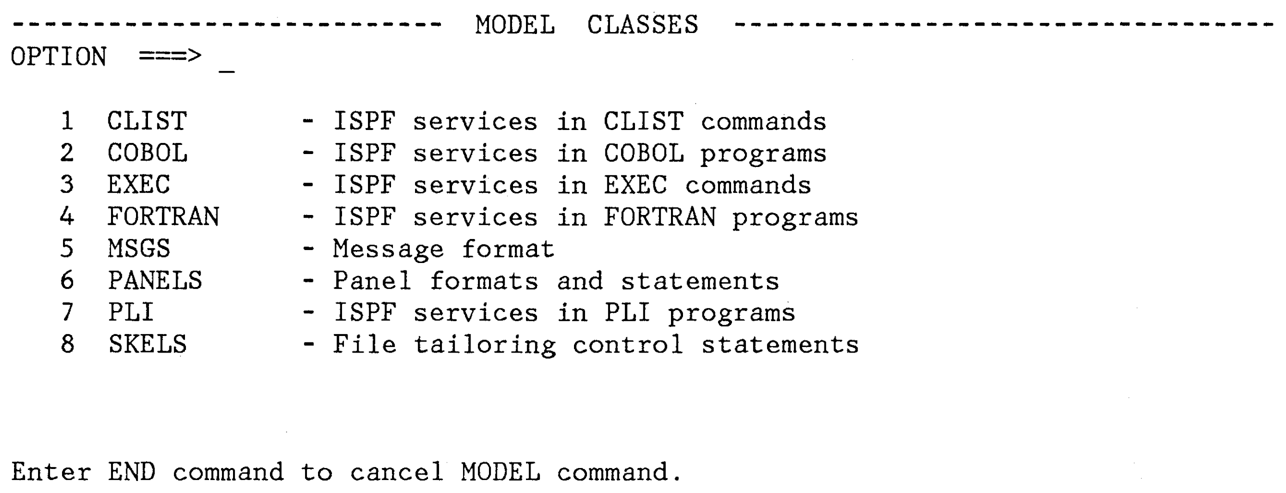
The second part of the logical name is the model **name**, which identifies a particular model within the model class. Frequently, it uniquely identifies a model and completes the logical name.

Optional parts of the logical name, called qualifiers, may be defined as needed to uniquely identify a model. Qualifiers are used, for example, to differentiate among the various kinds of panel verification (VER) statements.

This hierarchy of models is defined by a hierarchy of selection panels. The different parts of the logical name of a model are selections on the panels, which you may choose either by keyword name or option identifier. This allows you to be prompted by selection panels if you do not know the logical name of the model you want, or to bypass the display of these panels, if you do know the name.

To allow the model class to default whenever possible, that selection, if needed, is a separate interaction, as described below.

Typically, you need not be concerned with the model class. You must specify it only if you wish to use a class that is different from the 'type' qualifier of the data set you're editing. The 'type' qualifier



```
-------------------------- MODEL  CLASSES ---------------------------------
OPTION  ===> _

   1  CLIST        - ISPF services in CLIST commands
   2  COBOL        - ISPF services in COBOL programs
   3  EXEC         - ISPF services in EXEC commands
   4  FORTRAN      - ISPF services in FORTRAN programs
   5  MSGS         - Message format
   6  PANELS       - Panel formats and statements
   7  PLI          - ISPF services in PLI programs
   8  SKELS        - File tailoring control statements


 Enter END command to cancel MODEL command.
```

Figure 47. Edit - Model Class Selection Panel

is the PDF library type or the last qualifier in the file name. For example, you normally store panels in files whose type qualifier is PANELS. The model function of edit recognizes PANELS as a valid qualifier for panel models, so you do not need to specify the class when requesting a panel model from a file with a type qualifier of PANELS. Assume, however, that you call your panels "screens" and maintain them in a file with a type of SCREENS. When you wish to use a model to develop a new panel, and enter the MODEL command, SCREENS is not recognized as a model class, so you are prompted to identify the class you desire (in this case, the PANELS class).

Once you have set a class, whether by panel selection or by use of the MODEL CLASS command, that class remains in effect until changed, either until you enter a data set name with a different type qualifier, or until you leave the edit entry panel.

MODEL Command

Models are brought into the file being edited by the MODEL command. The MODEL command allows you to specify either:

- The logical name of the model to be copied
- The class of models to be searched when retrieving models

The command has the following format:

```
  MODEL [model-name [.qualifier...]]
        [CLASS [class-name]        ]
```

The operands are optional. One or more qualifiers following a model name may be required to uniquely identify a model; however, the qualifiers may not be used without a model name. Each additional qualifier after the model name implies another level in the hierarchy of model selection panels. If you omit the class name or model name, if you omit required qualifiers, or if there is a validation error, you are prompted with a selection panel from which you can select the desired class name, model name, or qualifier.

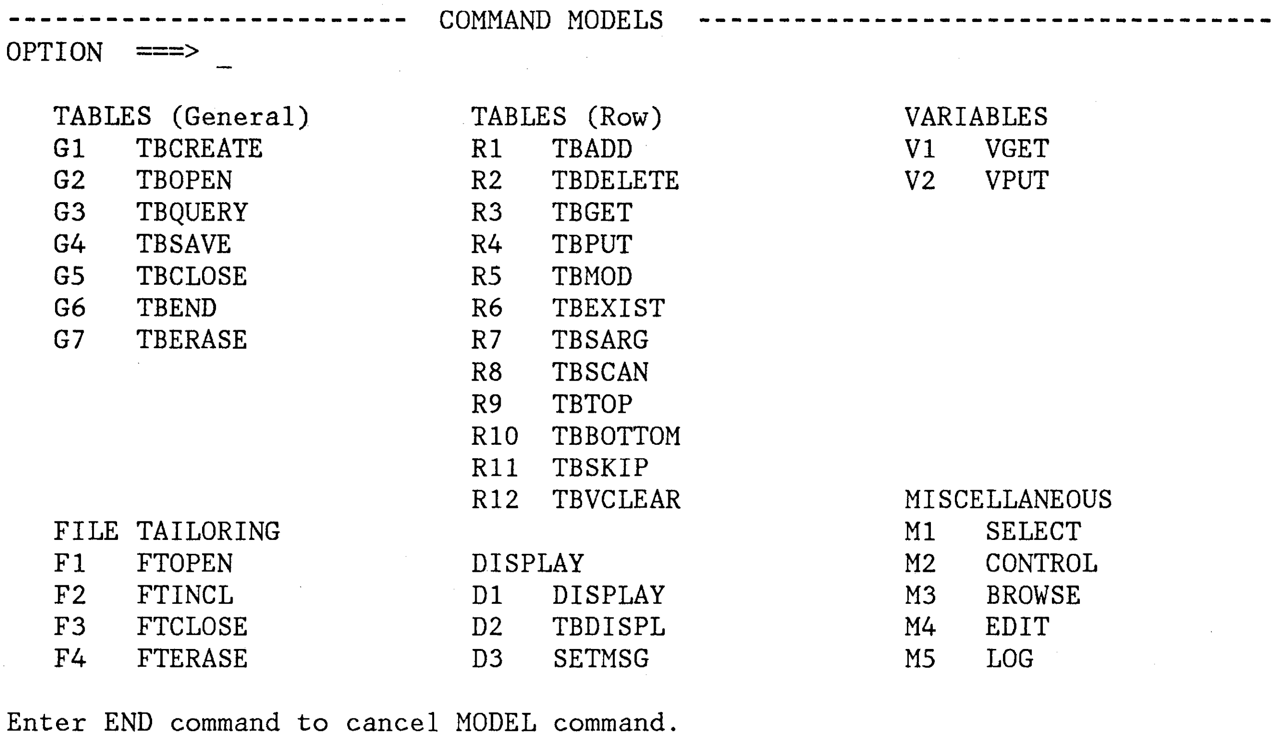
The class-name is optional and is used to specify the class of models to be used for the current edit session. If you specify a class name, it must be a name specified on the model class selection panel. If you omit it or provide an invalid class-name, the model class selection panel is displayed from which you can select a model class. Do not create a model with the model-name of "class" since this is a reserved word on the MODEL command.

Using Models

Models are used when you are editing a file and want assistance in defining a dialog element.

To use a model, first edit your file. Then, determine where you wish to place the model. If you are editing an existing file, use the A (after) or B (before) line command to indicate where the model will go. You need not use the A or B command when you have a new file. Then, enter the MODEL command in the command/option area (line 2) of the display. If you know the logical name of the model you want, enter it with the MODEL command. If you enter the MODEL command with no operands, you are presented with a series of selection panels, on which you indicate the desired model name and any desired qualifiers.

You are again presented with a display of your data, with the model in place. Now, you can overtype or use line commands to change the source statements in the model to meet the needs of your application.



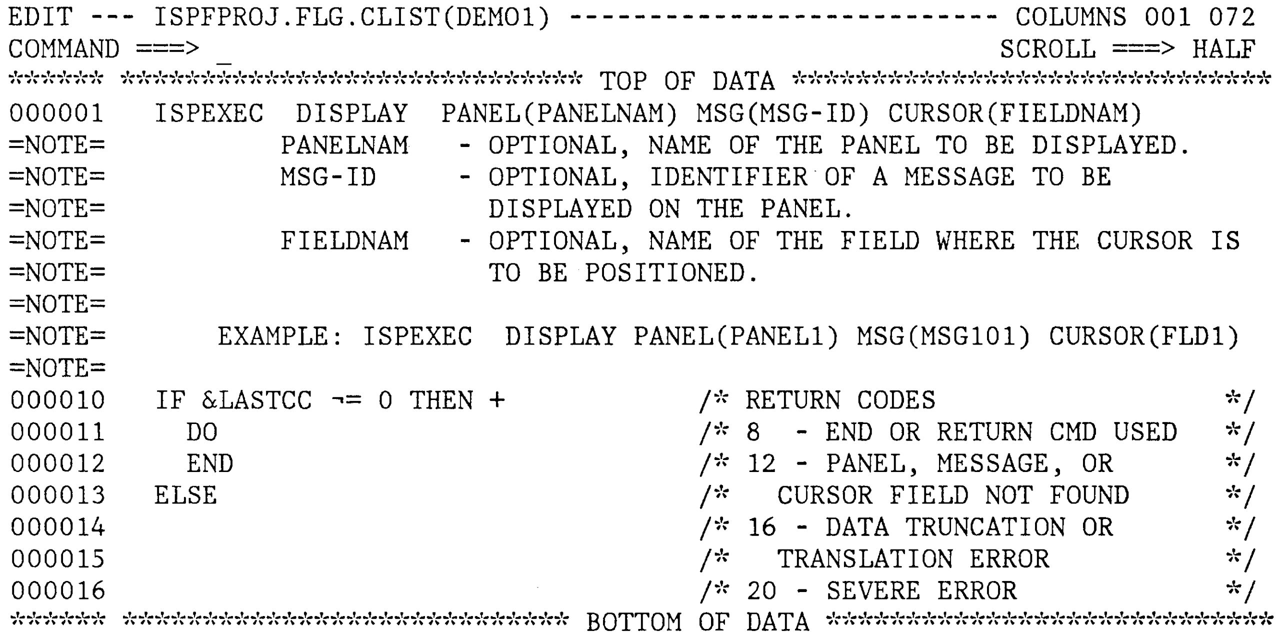
```
------------------------ COMMAND MODELS ---------------------------------
OPTION  ===> _

   TABLES (General)          TABLES (Row)              VARIABLES
   G1   TBCREATE             R1   TBADD                V1   VGET
   G2   TBOPEN               R2   TBDELETE             V2   VPUT
   G3   TBQUERY              R3   TBGET
   G4   TBSAVE               R4   TBPUT
   G5   TBCLOSE              R5   TBMOD
   G6   TBEND                R6   TBEXIST
   G7   TBERASE              R7   TBSARG
                             R8   TBSCAN
                             R9   TBTOP
                             R10  TBBOTTOM
                             R11  TBSKIP
                             R12  TBVCLEAR             MISCELLANEOUS
   FILE TAILORING                                      M1   SELECT
   F1   FTOPEN               DISPLAY                   M2   CONTROL
   F2   FTINCL               D1   DISPLAY              M3   BROWSE
   F3   FTCLOSE              D2   TBDISPL              M4   EDIT
   F4   FTERASE              D3   SETMSG               M5   LOG

Enter END command to cancel MODEL command.
```

Figure 48. Edit - Model Selection Panel for Command Models

As an example, assume that you are writing a dialog function using EXEC2 statements, and you want to have the EXEC display a panel. You are editing your EXEC member, called ISPFPROJ.FLG.EXEC(DEMO1). Since your file type (EXEC) matches the class of models you want, you can let the model class default. If you enter the MODEL command without a model name, the command models selection panel is displayed (Figure 48).

If you then select option D1 (DISPLAY), the model for the display service (Figure 49) is inserted in your EXEC at the location you've indicated with an A or B line command. Notice that any notes are identified by the characters "=NOTE=" in the line command area.

You can then continue editing the EXEC to change the display service call parameters for your function, using the notes as a guide for what to put in. The error handling code shown serves as a skeleton that you can update with your error-correction or recovery code. Finally, you can use the RESET command to eliminate the notes from the display, leaving the service call, the error-handling logic, and the comments.



```
EDIT --- ISPFPROJ.FLG.CLIST(DEMO1) -------------------------- COLUMNS 001 072
COMMAND ===> _                                                SCROLL ===> HALF
****** ********************************* TOP OF DATA **********************************
000001   ISPEXEC  DISPLAY  PANEL(PANELNAM) MSG(MSG-ID) CURSOR(FIELDNAM)
=NOTE=           PANELNAM   - OPTIONAL, NAME OF THE PANEL TO BE DISPLAYED.
=NOTE=           MSG-ID     - OPTIONAL, IDENTIFIER OF A MESSAGE TO BE
=NOTE=                        DISPLAYED ON THE PANEL.
=NOTE=           FIELDNAM   - OPTIONAL, NAME OF THE FIELD WHERE THE CURSOR IS
=NOTE=                        TO BE POSITIONED.
=NOTE=
=NOTE=       EXAMPLE: ISPEXEC  DISPLAY PANEL(PANEL1) MSG(MSG101) CURSOR(FLD1)
=NOTE=
000010   IF &LASTCC ¬= 0 THEN +           /* RETURN CODES                    */
000011     DO                             /* 8  - END OR RETURN CMD USED     */
000012     END                            /* 12 - PANEL, MESSAGE, OR         */
000013   ELSE                             /*   CURSOR FIELD NOT FOUND        */
000014                                    /* 16 - DATA TRUNCATION OR         */
000015                                    /*   TRANSLATION ERROR             */
000016                                    /* 20 - SEVERE ERROR               */
****** ****************************** BOTTOM OF DATA ********************************
```

Figure 49. Edit - Sample Dialog Development Model

Adding, Changing, and Deleting Models

Models are implemented in a very general fashion, so your installation can apply and use the concept for other tasks beside dialog development. You may want to create a set of DL/I call models for your IMS applications, or a set of report format models for your sales forecasting application. You can also create models for the "canned" JCL statements that you use most frequently.

Similarly, you may find that the models provided for panel formats do not correspond to the standards for your local installation or for your particular application. You can modify the distributed panel models to match your own requirements.

This section describes how you can add a new model to your skeleton library, modify an existing model, or delete an existing model.

Adding Models

To create a new model, you must:

1. Determine the file name and member name for the model. For actual use, the model must be in a skeleton library.

2. Create the source code for the model. Consider whether you should create all new code or whether you should modify an existing model under a new name.

3. Make the model accessible from a model selection panel by having its selection call the program ISRECMBR with the actual model member name as its parameter. This can involve:

   - Modifying an existing model selection panel to add the new panel.

   - Creating a new model selection panel. If you do this, you must add the new panel to the hierarchy of selection panels by changing one of the higher level panels.

   - No change, if you are replacing an existing model with an updated model with the same name.

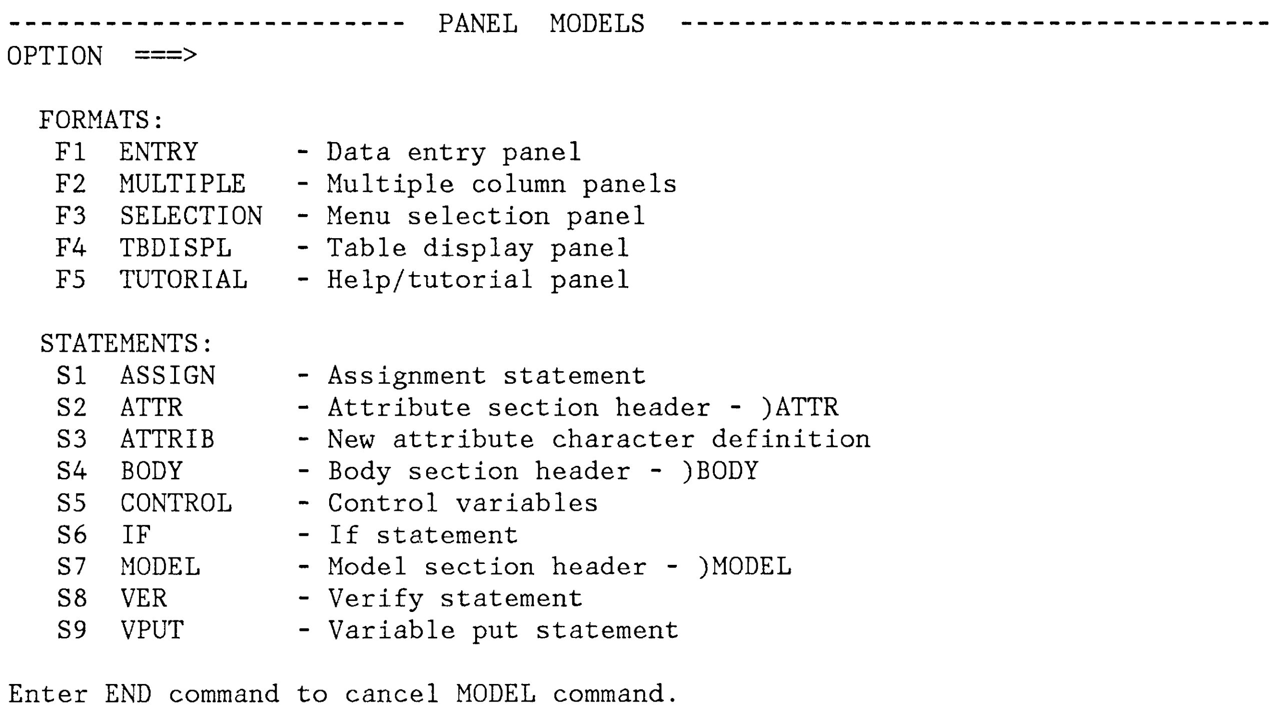
As an example of adding a model, let us assume that you wish to create a model for multi-line block letters. Since you intend to use these block letters on panels, the model will become part of the panel model class.

To build a model block letter, use the edit function to create a new member in your skeleton library (for example, BLKI). By manipulating input, you might develop the letter "I" that appears in Figure 50.

```
        IIIIIIIIII
            II
            II
            II
            II
            II
        IIIIIIIIII
)N
)N   the letter I for logo
```

Figure 50. Edit - Sample Block Letter Model

Once the model for each letter is built, you must update the selection panel in the prompting sequence that deals with panel model selection. The displayed form of this panel (ISREMPNL in the system panel library) is shown in Figure 51.

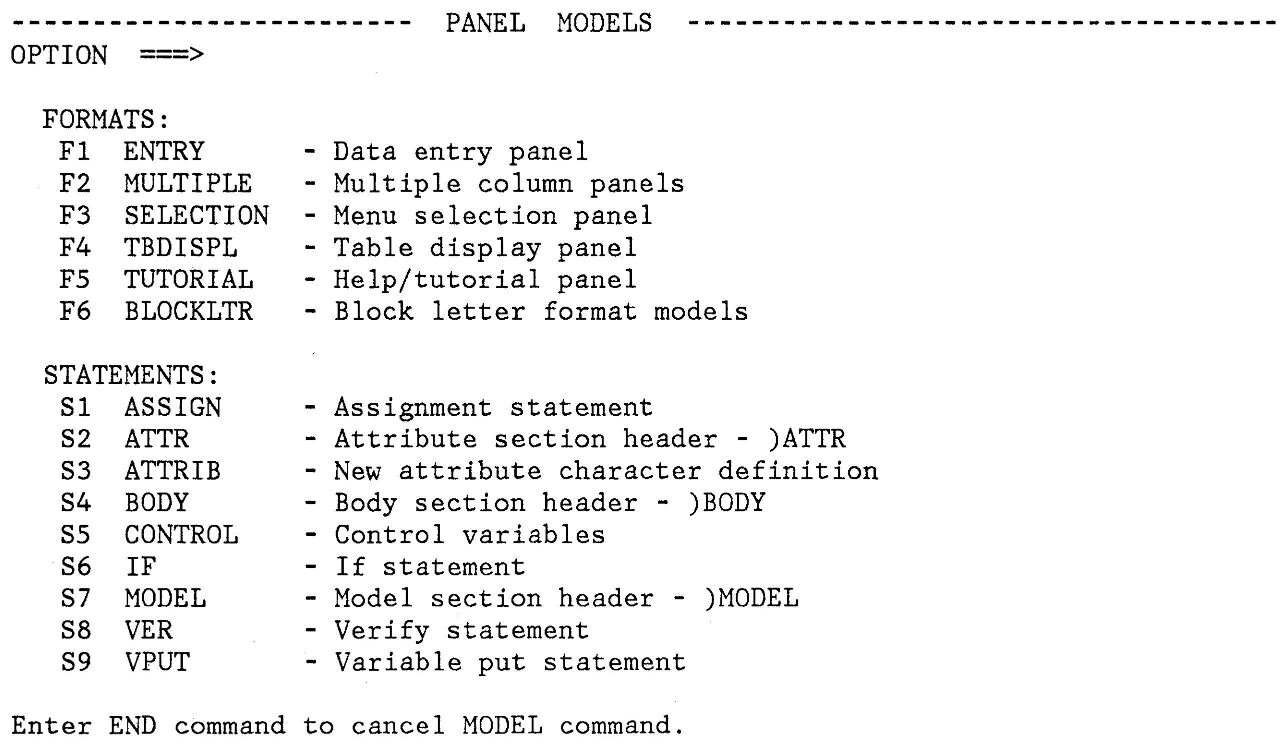


```
------------------------ PANEL MODELS ----------------------------------
OPTION ===>

  FORMATS:
   F1  ENTRY     - Data entry panel
   F2  MULTIPLE  - Multiple column panels
   F3  SELECTION - Menu selection panel
   F4  TBDISPL   - Table display panel
   F5  TUTORIAL  - Help/tutorial panel

  STATEMENTS:
   S1  ASSIGN    - Assignment statement
   S2  ATTR      - Attribute section header - )ATTR
   S3  ATTRIB    - New attribute character definition
   S4  BODY      - Body section header - )BODY
   S5  CONTROL   - Control variables
   S6  IF        - If statement
   S7  MODEL     - Model section header - )MODEL
   S8  VER       - Verify statement
   S9  VPUT      - Variable put statement

 Enter END command to cancel MODEL command.
```

Figure 51. Edit - Panel Model Selection Panel

Copy this panel into your panel data set and modify it, as shown in Figure 52, by adding a new format, F6. Note the addition of Format F6, BLOCKLTR.

Since we are assuming there are several new models, this panel should be updated so that when a user selects F6, a new block letters selection panel should be displayed. Therefore, you should modify the)PROC section of panel ISREMPNL to include item F6, as shown in Figure 53.



```
------------------------- PANEL MODELS -------------------------------------
OPTION ===>

  FORMATS:
   F1  ENTRY      - Data entry panel
   F2  MULTIPLE   - Multiple column panels
   F3  SELECTION  - Menu selection panel
   F4  TBDISPL    - Table display panel
   F5  TUTORIAL   - Help/tutorial panel
   F6  BLOCKLTR   - Block letter format models

  STATEMENTS:
   S1  ASSIGN     - Assignment statement
   S2  ATTR       - Attribute section header - )ATTR
   S3  ATTRIB     - New attribute character definition
   S4  BODY       - Body section header - )BODY
   S5  CONTROL    - Control variables
   S6  IF         - If statement
   S7  MODEL      - Model section header - )MODEL
   S8  VER        - Verify statement
   S9  VPUT       - Variable put statement

Enter END command to cancel MODEL command.
```

Figure 52. Edit - Modified Visible Panel Model Selection Panel

```
)PROC
  IF (&ZCMD = 'SELECTION')
      &TMP = TRUNC (&ZCMD,'.')
      &ZCMD = TRUNC (&ZCMD,8)
  &ZSEL = TRANS(TRUNC (&ZCMD,'.')
        F1,'PGM(ISRECMBR) PARM(ISREMMF1)'
     ENTRY,'PGM(ISRECMBR) PARM(ISREMMF1)'
        F2,'PANEL(ISREMSEL)'                /*  NEED TO QUALIFY THIS  */
  MULTIPLE,'PANEL(ISREMSEL)'                /*  PANEL FOR COLUMNS ID. */
        F3,'PGM(ISRECMBR) PARM(ISREMMF3)'
    SELECT,'PGM(ISRECMBR) PARM(ISREMMF3)'   /* AUTOMATIC SINGLE COLUMN*/
  SELECTIO,'PGM(ISRECMBR) PARM(ISREMMF3)'   /* FOR 8 OR LESS SELECTION*/
        F4,'PGM(ISRECMBR) PARM(ISREMMF4)'
   TBDISPL,'PGM(ISRECMBR) PARM(ISREMMF4)'
        F5,'PGM(ISRECMBR) PARM(ISREMMF5)'
  TUTORIAL,'PGM(ISRECMBR) PARM(ISREMMF5)'
        F6,'PANEL(BLKLTRS)'                 /* LARGE BLOCK LETTERS    */
  BLOCKLTR,'PANEL(BLKLTRS)'
        S1,'PANEL(ISREMASG)'                /*  ASSIGN AND VERIFY     */
    ASSIGN,'PANEL(ISREMASG)'                /*    NEED QUALIFIERS     */
   etc. etc. etc.
```

Figure 53. Edit - Modified)PROC Section of Panel Model Selection Panel

Note that the entry for F6, BLOCKLTR, points to a new panel, BLKLTRS, which you must now build.

You may wish to modify an existing panel model to create the new panel, which might appear as shown in Figure 54. Note particularly the)INIT and)PROC sections of the coding.

In the)PROC section of panel BLKLTRS, the target for all valid selections is the program ISRECMBR. The parameter passed to this program is unique for each unique valid selection and is the name of the model for that selection. Thus, for our example, the model name for selection 1 or I is BLKI.

You should follow the)INIT code and the end code in the)PROC section shown in Figure 54 for all new panels.

```
)ATTR
)BODY
%------------------------- BLOCK LETTER ---------------------------------------
%OPTION  ===>_ZCMD

%
%   1 +I           - Block letter I
%   2 +J           - Block letter J
%   3 +K           - Block letter K
%
%
+Enter%END+command to cancel MODEL command.

%
)INIT
  .CURSOR = ZCMD
  .HELP = ISRxxxxx
  IF (&ISRMDSPL = 'RETURN  ')
       .RESP = END
)PROC
  &ZSEL = TRANS(TRUNC (&ZCMD,'.')
         1,'PGM(ISRECMBR) PARM(BLKI)'
         I,'PGM(ISRECMBR) PARM(BLKI)'
         2,'PGM(ISRECMBR) PARM(BLKJ)'
         J,'PGM(ISRECMBR) PARM(BLKJ)'
         3,'PGM(ISRECMBR) PARM(BLKK)'
         K,'PGM(ISRECMBR) PARM(BLKK)'
         *,'?' )
  IF (&ZSEL = '?')
     .MSG = ISRYM012
  &ISRMMEND = 'N'                   /* SET THE END INDICATOR TO NO   */
  IF (.RESP = END)                  /* IF ENDING, WHY ... WHO CAUSED */
    IF (&ISRMONCL ¬= 'Y')           /* MAKE SURE ITS NOT A CLASS OP. */
      IF (&ISRMDSPL ¬= 'RETURN  ')  /* MAKE SURE ITS NOT END ON MBR. */
        &ISRMMEND = 'Y'             /* NO - ITS BECAUSE USER HIT END */
)END
```

Figure 54. Edit - Block Letter Model Selection Panel Coding

Finding Models

Before you change or delete a model, you must determine the physical name of the model in the skeleton library. The names of the models of dialog elements distributed with PDF are shown in Appendix G. You can use the following method to find the member name for any model.

You can find the member name for any model in the)PROC section of the final selection panel used to obtain it. The member name is the parameter passed to ISRECMBR, the program called when that selection is chosen by the user. In Figure 53, for example, the name of the model for a selection panel format is ISREMMF3.

To determine the name of the model selection panel so that you can look at it to find the model member name, use the PANELID command when that panel is displayed on the screen. Then use browse or edit to look at the member of the panel library with that name.

Changing Models

To modify a model that currently exists, copy the existing model from the skeleton data set into your own data set. Then use edit to change the model as you would change any text file. Note that any lines that are to contain notes must have ")N" in positions 1 and 2, as shown below.

```
    VARIABLE = VALUE
)N        VARIABLE - A DIALOG VARIABLE OR A CONTROL VARIABLE.
)N        VALUE    - A LITERAL VALUE CONTAINING:  SUBSTITUTABLE
)N                   VARIABLES, A DIALOG VARIABLE, A CONTROL VARIABLE,
)N                   OR AN EXPRESSION CONTAINING A BUILT-IN FUNCTION.
)N        EXAMPLES:  &DEPT = 'Z59'   &A = &B    &C = ' '
```

When the model is later accessed using the MODEL command, the lines with)N indicators are flagged with '=NOTE=' in the line command area (Figure 49).

Deleting Models

You can delete models by deleting the references to them. To delete the references, remove the entry referencing the model in both the)BODY and)PROC sections of the model selection panel.

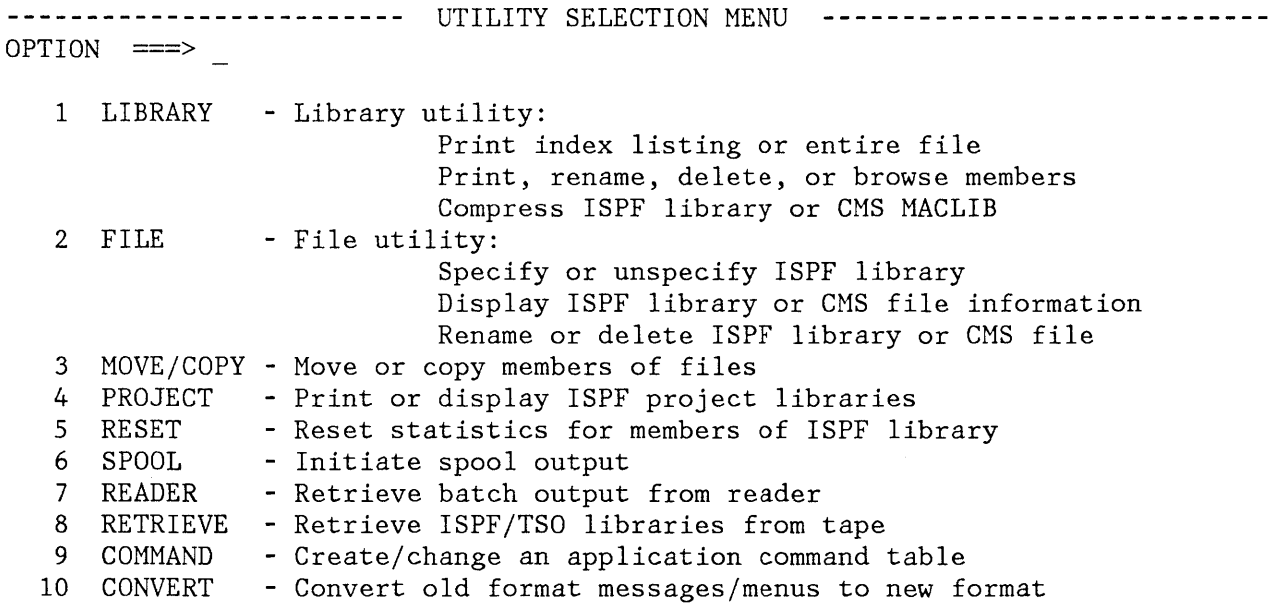
Generally, you can leave the model itself in the skeleton library. However, if you are deleting a substantial number of models, you may wish to consider deleting those members from the library and compressing the library.

CHAPTER 7. UTILITIES (OPTION 3)

The utility option provides a variety of functions for library and file maintenance, moving and copying data, printing or displaying PDF project listings, resetting library statistics, initiating spool output, retrieving data from the virtual reader, retrieving ISPF libraries (via tape) from a TSO system, creating and modifying application command tables, and converting old format messages and menus to new format.

The utility selection panel is shown in Figure 55.

The following sections describe each of the utility functions.



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

   1  LIBRARY    - Library utility:
                           Print index listing or entire file
                           Print, rename, delete, or browse members
                           Compress ISPF library or CMS MACLIB
   2  FILE       - File utility:
                           Specify or unspecify ISPF library
                           Display ISPF library or CMS file information
                           Rename or delete ISPF library or CMS file
   3  MOVE/COPY  - Move or copy members of files
   4  PROJECT    - Print or display ISPF project libraries
   5  RESET      - Reset statistics for members of ISPF library
   6  SPOOL      - Initiate spool output
   7  READER     - Retrieve batch output from reader
   8  RETRIEVE   - Retrieve ISPF/TSO libraries from tape
   9  COMMAND    - Create/change an application command table
  10  CONVERT    - Convert old format messages/menus to new format
```

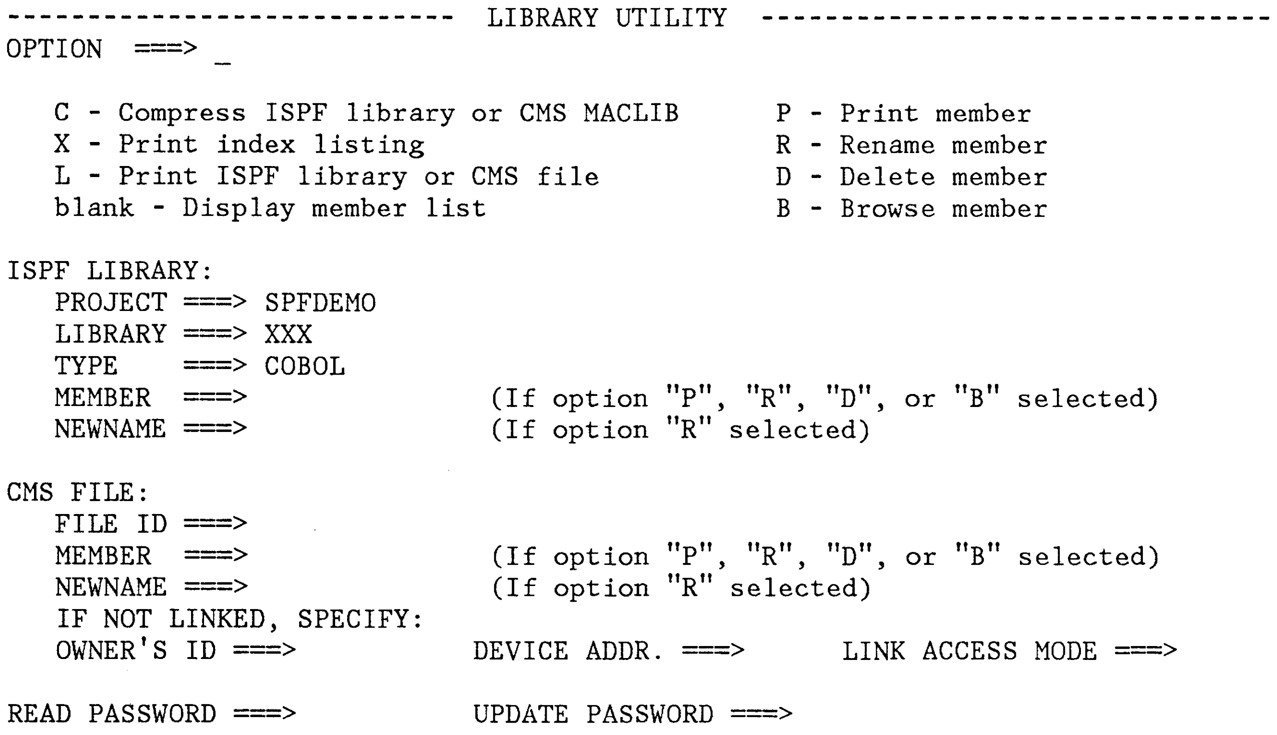
Figure 55. Utility Selection Panel

LIBRARY UTILITY (OPTION 3.1)

When this option is selected, a panel is displayed that allows you to specify a library or file and indicate an action to be performed (Figure 56). Possible actions are:

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

The library utility is intended primarily for maintenance of ISPF libraries, MACLIBs, and TXTLIBs, but the print index listing function (option X) and the print ISPF library or CMS file function (option L) also apply to sequential CMS files.



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

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

ISPF 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 56. Library Utility Panel

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

If you select option X (print index listing), an ISPF library or CMS file may be specified. The index listing is recorded in the ISPF list file. For ISPF libraries, MACLIBs, 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 the index listing format.

If you select option L (print ISPF library or CMS file), a 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 ISPF list file.

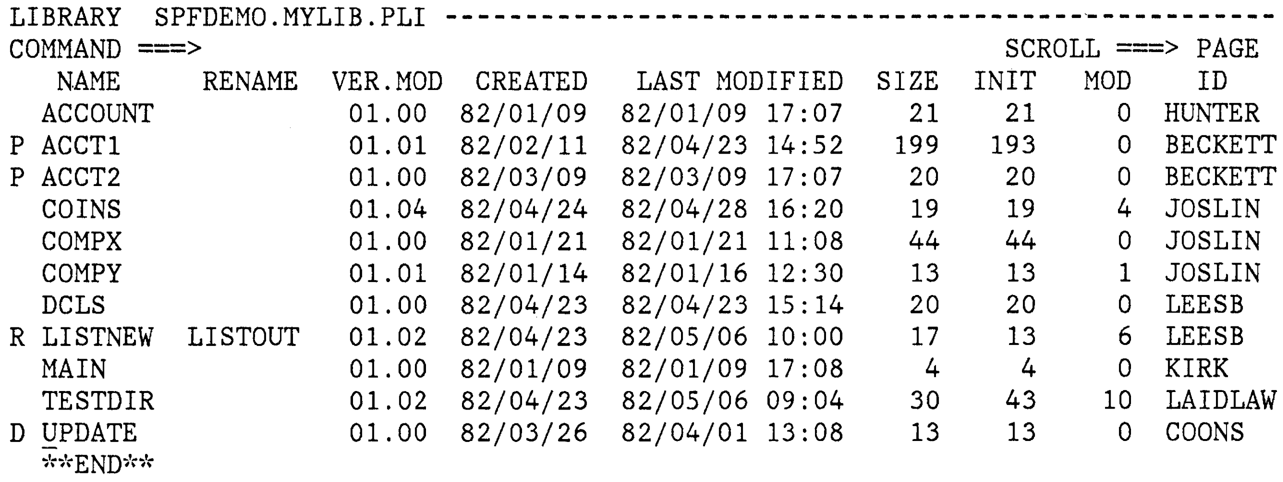
If you select option P (print member), a member of an ISPF library, MACLIB, or TXTLIB may be specified. A source listing of the member is recorded in the ISPF list file.

If you select option R or option D (rename member or delete member), a member of an ISPF library, MACLIB, or TXTLIB may be specified. A new member name must also be specified for the "rename member" function.

If you select option B (browse member), a member of an ISPF library, MACLIB, or TXTLIB may be specified. The specified member is displayed in browse mode. You can use all the browse commands. When you terminate browse by entering the END command, the library utility panel is redisplayed.

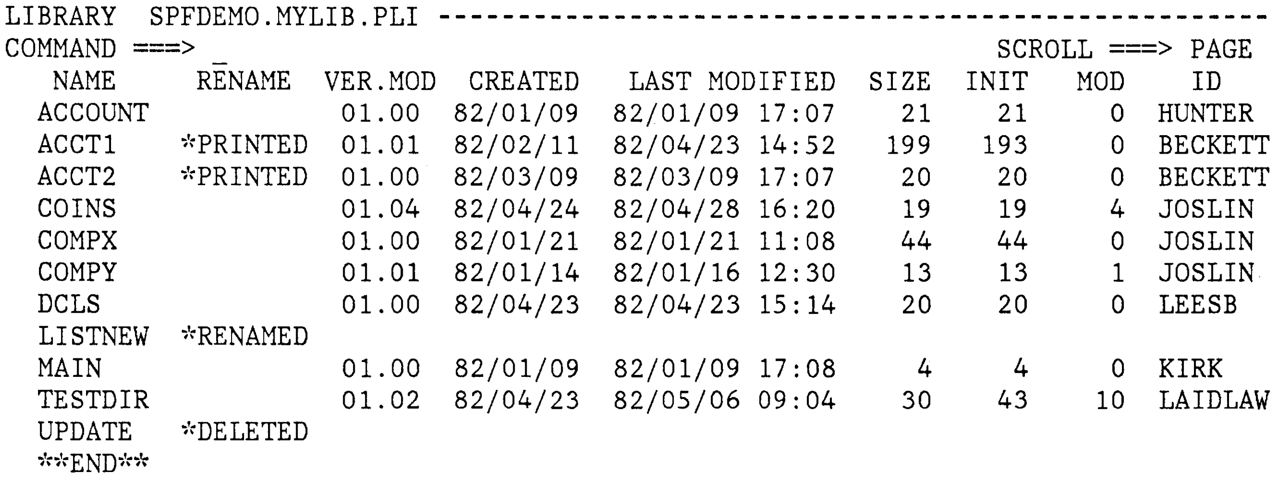
If you leave the option field blank (display a member list), you may specify an ISPF library, MACLIB, or TXTLIB. A member list is then displayed from which you may select members for printing, renaming, deleting, or browsing by entering "P", "R", "D", or "B", respectively, in front of one or more member names. For renaming, you must also enter a new member name in the field immediately following the current member name. You can scroll the member list up and down using the scroll commands or the LOCATE command. You terminate the member list by entering the END command.

Figure 57 shows a before-and-after example where members ACCT1 and ACCT2 are printed, member UPDATE is deleted, and member LISTNEW is renamed LISTOUT.



```
LIBRARY  SPFDEMO.MYLIB.PLI -----------------------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE
   NAME     RENAME   VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT             01.00  82/01/09  82/01/09 17:07    21    21     0  HUNTER
P ACCT1               01.01  82/02/11  82/04/23 14:52   199   193     0  BECKETT
P ACCT2               01.00  82/03/09  82/03/09 17:07    20    20     0  BECKETT
  COINS               01.04  82/04/24  82/04/28 16:20    19    19     4  JOSLIN
  COMPX               01.00  82/01/21  82/01/21 11:08    44    44     0  JOSLIN
  COMPY               01.01  82/01/14  82/01/16 12:30    13    13     1  JOSLIN
  DCLS                01.00  82/04/23  82/04/23 15:14    20    20     0  LEESB
R LISTNEW  LISTOUT    01.02  82/04/23  82/05/06 10:00    17    13     6  LEESB
  MAIN                01.00  82/01/09  82/01/09 17:08     4     4     0  KIRK
  TESTDIR             01.02  82/04/23  82/05/06 09:04    30    43    10  LAIDLAW
D UPDATE              01.00  82/03/26  82/04/01 13:08    13    13     0  COONS
  **END**
```

Figure 57 (Part 1 of 2). Library Utility - Print, Rename, and Delete



```
LIBRARY  SPFDEMO.MYLIB.PLI --------------------------------------------------
COMMAND ===> _                                              SCROLL ===> PAGE
   NAME      RENAME   VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT   MOD    ID
  ACCOUNT             01.00  82/01/09  82/01/09 17:07    21    21     0  HUNTER
  ACCT1     *PRINTED  01.01  82/02/11  82/04/23 14:52   199   193     0  BECKETT
  ACCT2     *PRINTED  01.00  82/03/09  82/03/09 17:07    20    20     0  BECKETT
  COINS               01.04  82/04/24  82/04/28 16:20    19    19     4  JOSLIN
  COMPX               01.00  82/01/21  82/01/21 11:08    44    44     0  JOSLIN
  COMPY               01.01  82/01/14  82/01/16 12:30    13    13     1  JOSLIN
  DCLS                01.00  82/04/23  82/04/23 15:14    20    20     0  LEESB
  LISTNEW   *RENAMED
  MAIN                01.00  82/01/09  82/01/09 17:08     4     4     0  KIRK
  TESTDIR             01.02  82/04/23  82/05/06 09:04    30    43    10  LAIDLAW
  UPDATE    *DELETED
  **END**
```

Figure 57 (Part 2 of 2). Library Utility - Print, Rename, and Delete

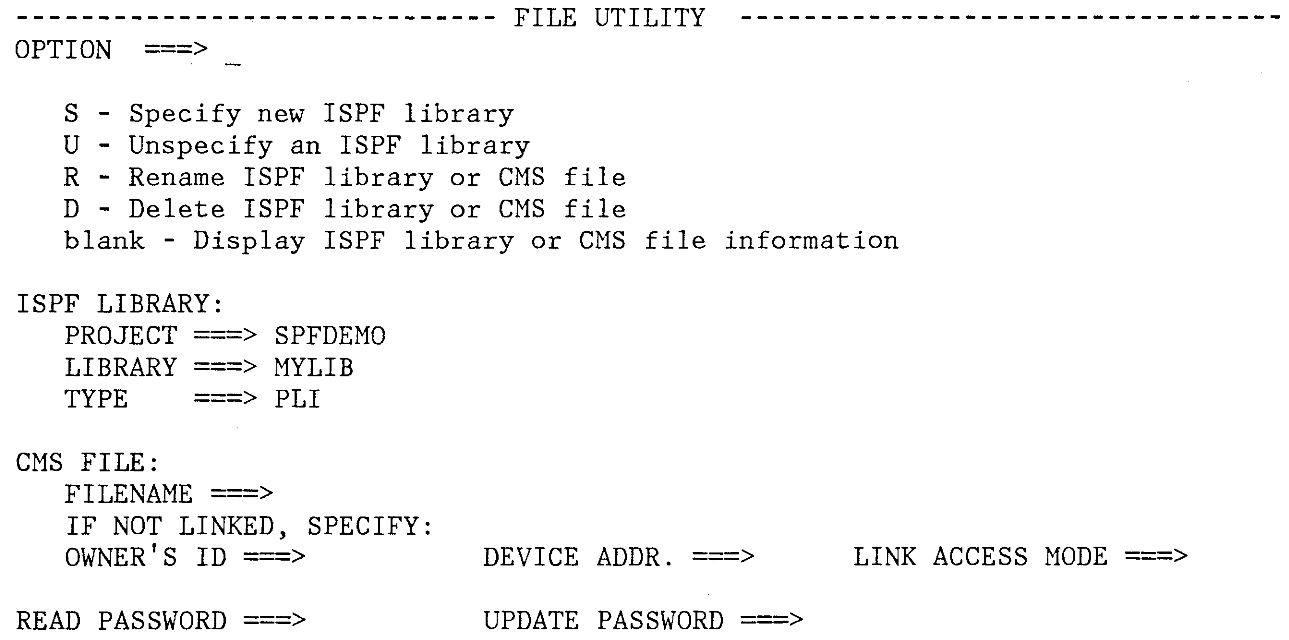
]

FILE UTILITY (OPTION 3.2)

When you select this option, a panel is displayed that allows you to specify a library or file and indicate an action to be performed (Figure 58). Possible actions are:

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

For options S and U, you must specify a library identification (project, library, and type). For the other options, you may specify either an ISPF library or CMS file.



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

   S - Specify new ISPF library
   U - Unspecify an ISPF library
   R - Rename ISPF library or CMS file
   D - Delete ISPF library or CMS file
   blank - Display ISPF library or CMS file information

ISPF 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 ===>
```

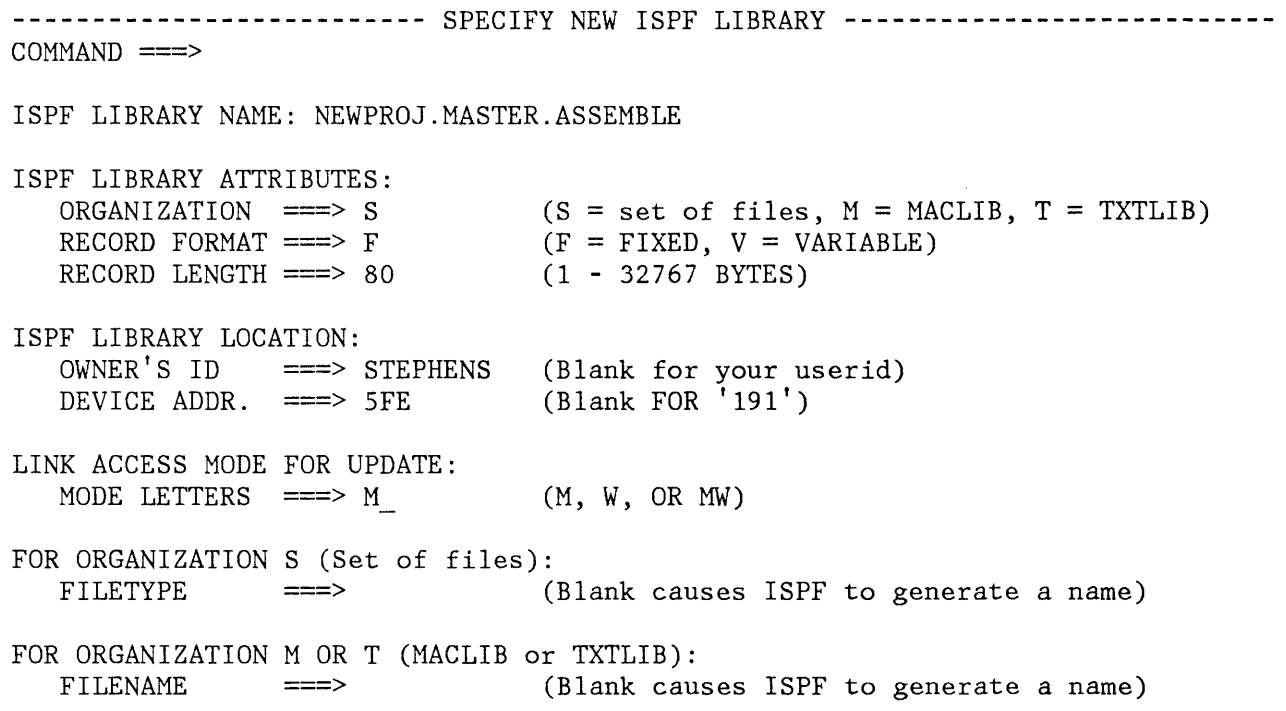
Figure 58. File Utility Panel

If you select option S (specify new ISPF library), a panel is displayed to allow you to specify library attributes and location (minidisk), link access mode for update, and corresponding CMS filetype/filename (Figure 59). Default values are pre-entered based on what you last entered on this panel, or based on the last display file information request (whichever occurred most recently). You may overtype the displayed defaults. When you press ENTER, the new library is specified. Entering the END command returns you to the previous panel without performing the library specification.

If the specified library location is not your A-disk (191), PDF attempts to access the minidisk using the designated link access mode (M, W, or MW) for update. This link fails if you specified mode W and the minidisk is currently linked.

See Appendix B for guidelines in specifying ISPF libraries.

If you select option U (unspecify ISPF library), a confirmation panel is displayed to ensure that you did not select this option by mistake. As directed in the panel, press either ENTER to confirm or enter the END command to cancel. Either action causes a return to the previous panel.



```
------------------------- SPECIFY NEW ISPF LIBRARY ---------------------------
COMMAND ===>

ISPF LIBRARY NAME: NEWPROJ.MASTER.ASSEMBLE

ISPF 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)

ISPF 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 ISPF to generate a name)

FOR ORGANIZATION M OR T (MACLIB or TXTLIB):
   FILENAME      ===>            (Blank causes ISPF to generate a name)
```

Figure 59. File Utility - Specify New ISPF Library

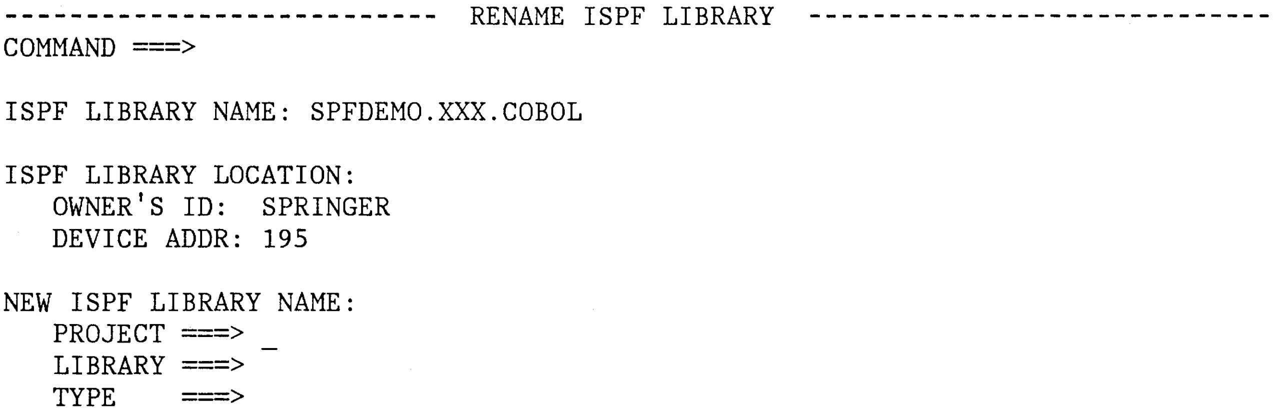
Option U causes the information about the library to be purged from the PDF catalog, but it does not cause the associated file(s) to be deleted.

If you select option R (rename file or ISPF library), a panel is displayed to allow you to enter the new file or library identification (Figure 60). Type the new identifier(s) and press ENTER to rename, or enter the END command to cancel. Either action causes a return to the previous panel.

If you select option D (delete file or ISPF library), a confirmation panel is displayed to ensure that you did not select this option by mistake. As directed in the panel, press either ENTER to confirm or enter the END command to cancel. Either action causes a return to the previous panel.

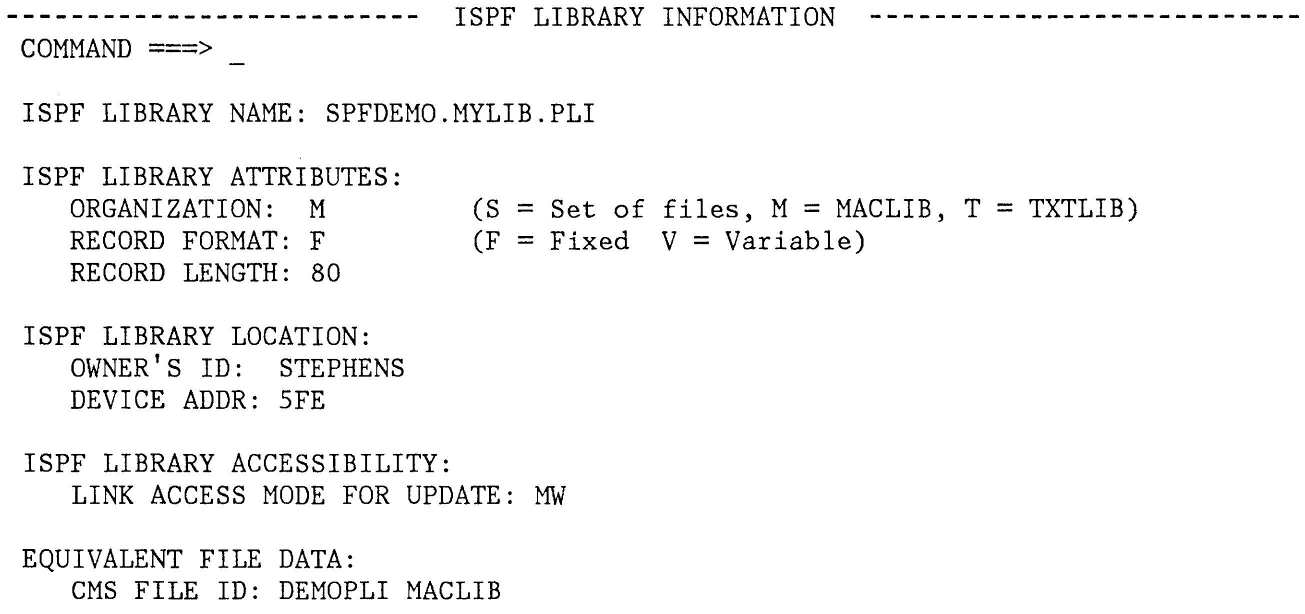
If you specify a library, option D causes the library to be unspecified and the associated file(s) to be deleted (erased).

If you leave the option field blank (to display file or ISPF library information), the location, characteristics, and current space utilization of the specified library or file are displayed (Figure 61). To return to the previous panel, press the ENTER key or enter the END command.



```
-------------------------- RENAME ISPF LIBRARY ----------------------------
COMMAND ===>

ISPF LIBRARY NAME: SPFDEMO.XXX.COBOL

ISPF LIBRARY LOCATION:
   OWNER'S ID:  SPRINGER
   DEVICE ADDR: 195

NEW ISPF LIBRARY NAME:
   PROJECT ===> _
   LIBRARY ===>
   TYPE    ===>
```

Figure 60. File Utility - Rename ISPF Library



```
------------------------- ISPF LIBRARY INFORMATION --------------------------
 COMMAND ===> _

 ISPF LIBRARY NAME: SPFDEMO.MYLIB.PLI

 ISPF LIBRARY ATTRIBUTES:
    ORGANIZATION:  M        (S = Set of files, M = MACLIB, T = TXTLIB)
    RECORD FORMAT: F        (F = Fixed  V = Variable)
    RECORD LENGTH: 80

 ISPF LIBRARY LOCATION:
    OWNER'S ID:  STEPHENS
    DEVICE ADDR: 5FE

 ISPF LIBRARY ACCESSIBILITY:
    LINK ACCESS MODE FOR UPDATE: MW

 EQUIVALENT FILE DATA:
    CMS FILE ID: DEMOPLI MACLIB
```

Figure 61. File Utility - Display Information

MOVE/COPY UTILITY (OPTION 3.3)

When you select this option, a panel is displayed that allows you to specify the "from" library or file (and member if it is an ISPF library, MACLIB, or TXTLIB), and to indicate an action to be performed (Figure 62). 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 you press ENTER, a second panel is displayed that allows you to specify the "to" library or file (Figure 62). You must also specify the following options on this panel:

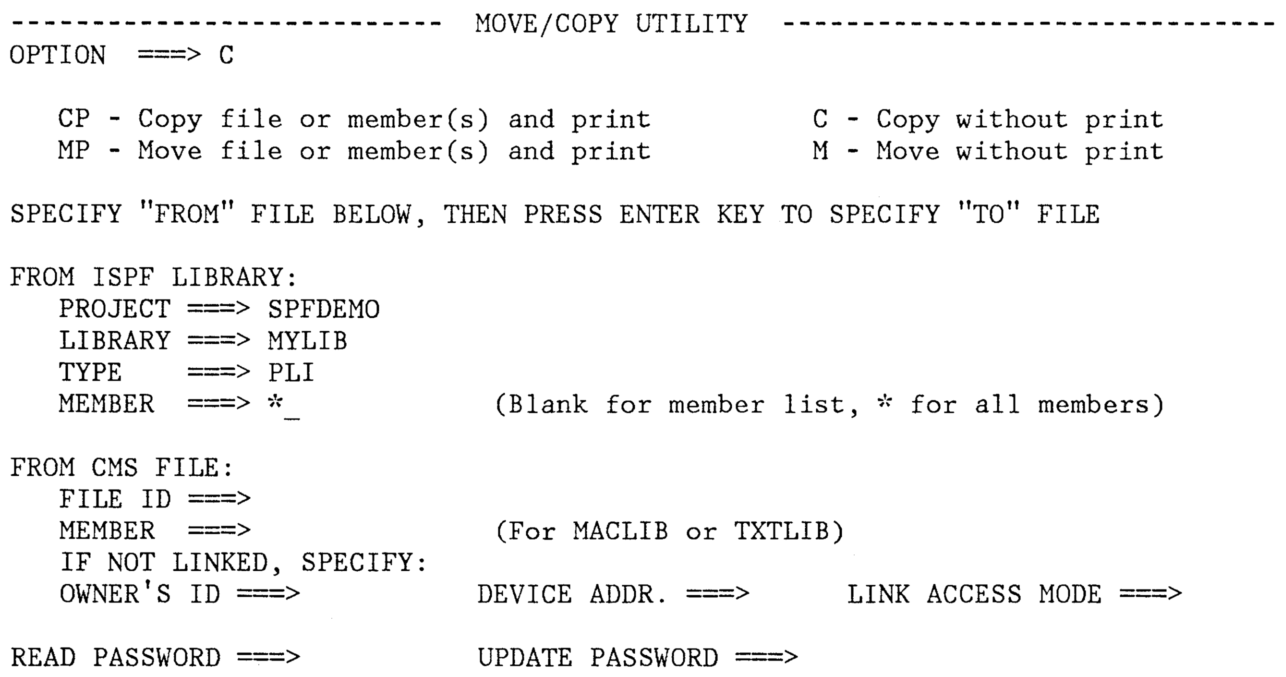
- Replace like-named members (YES or NO), if the "to" file is a library, MACLIB, or TXTLIB.

- "To" CMS file disposition (OLD or MOD), if the "to" file is sequential.

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.

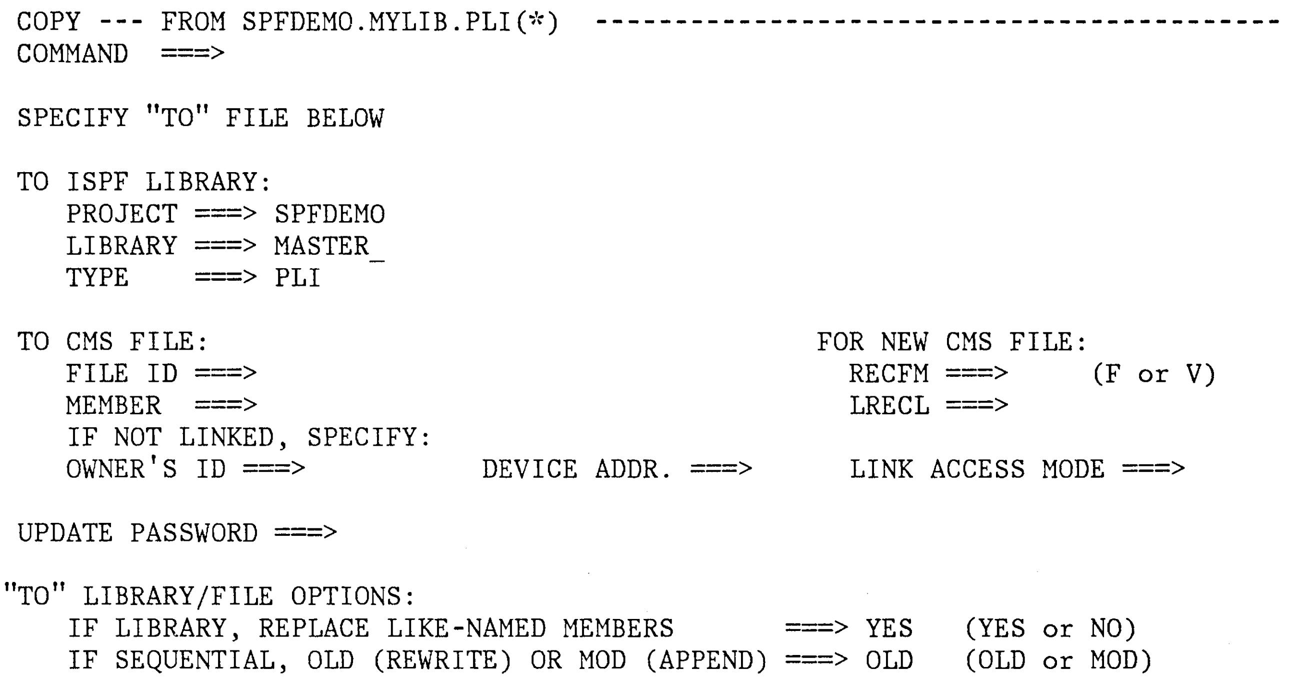
You can specify any library or CMS file for either the "from" or "to" file, but you may not specify the same library, MACLIB, or TXTLIB as both the "from" and "to" file.

If the "to" file is a CMS file which does not currently exist, a new file is created with the same attributes as the "from" file. If the "to" file is a library, it must have been previously specified via option 3.2 (specify ISPF library).



```
---------------------------- MOVE/COPY UTILITY -------------------------------
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 KEY TO SPECIFY "TO" FILE

FROM ISPF 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 ===>
```

Figure 62 (Part 1 of 2). Move/Copy Utility Panels



```
 COPY --- FROM SPFDEMO.MYLIB.PLI(*)  ------------------------------------------
 COMMAND  ===>

 SPECIFY "TO" FILE BELOW

 TO ISPF 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 ===>

"TO" LIBRARY/FILE OPTIONS:
    IF LIBRARY, REPLACE LIKE-NAMED MEMBERS       ===> YES   (YES or NO)
    IF SEQUENTIAL, OLD (REWRITE) OR MOD (APPEND) ===> OLD   (OLD or MOD)
```

Figure 62 (Part 2 of 2). Move/Copy Utility Panels

If you indicate M (move) and the "from" file is a library, MACLIB, or TXTLIB, the selected member(s) are deleted from the "from" file after they have been successfully copied to the "to" file. If you indicate M (move) and the "from" file is a sequential CMS file, the entire "from" file is deleted after its contents have been successfully copied to the "to" file. If you indicate C (copy), no deletions occur.

If you indicate either MP (move and print) or CP (copy and print), source listing(s) are recorded in the ISPF list file. If the "to" file is a library, MACLIB, or TXTLIB, a listing of each new or replaced member is recorded. If the "to" file is a sequential CMS file, a listing of its entire contents is recorded after the information has been successfully copied.

The following rules apply for specifying member names if the "from" file is a 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 list to be displayed (after the second panel has been displayed).

- Specifying a "from" member name of asterisk (*) causes all members to be moved or copied.

If the "to" file is a library or MACLIB, you must specify the "to" member name 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 a member list -- see below.)

For all other conditions, leave the "to" member blank.

A member list is displayed when the "from" file is a library, MACLIB, or TXTLIB, and the "from" member name is left blank. You 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 panel. If the "to" file is a library or MACLIB, you can also rename the member(s) by entering a new member name in the field immediately following the current name.

You may scroll the member list up and down using the scrolling commands or the LOCATE command. To terminate the member list, enter the END command, which causes a return to the first move/copy panel.

Figure 63 shows a before-and-after example in which 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).

- Multiple members of a library, MACLIB, or TXTLIB may be moved or copied to a sequential CMS file by using a member list or by specifying all members (*). The merged members appear in the "to" file, replacing the original contents of the file (if disposition OLD was specified) or added to the end (if disposition MOD was specified).

- If the "to" file is a 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).

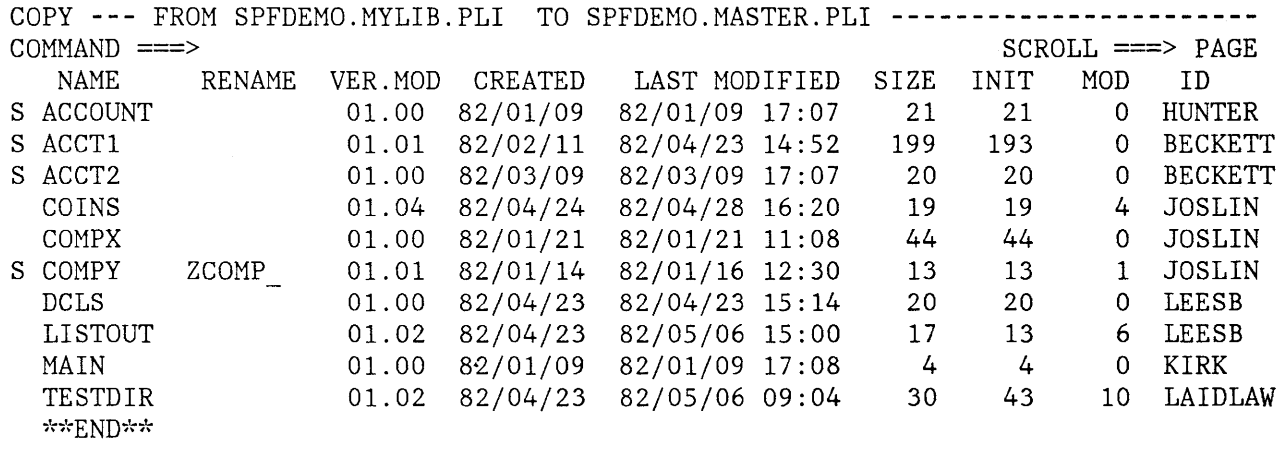
- Record formats and logical record lengths for the two files need not be alike. When necessary, data is truncated or right-padded with blanks to accommodate different record lengths.

- 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 data.

- Library statistics are not modified when moving or copying between libraries.

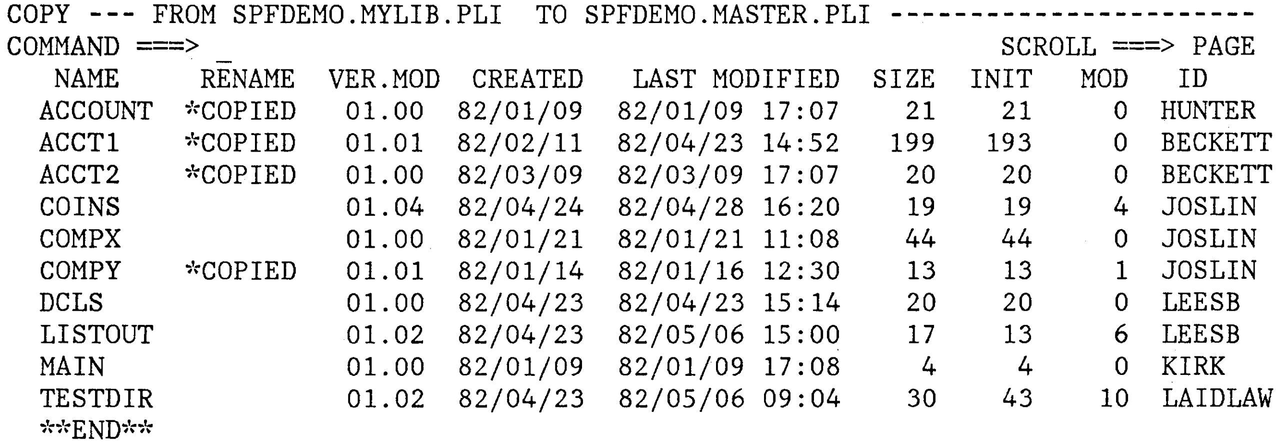
If the "to" file is a TXTLIB (either a library or a CMS file), PDF invokes the CMS TXTLIB command to update the file. The member name in the TXTLIB is determined by the TXTLIB command, based on information in the text module. You may not specify a member name (when moving or copying from a sequential TEXT file) or rename the member (when moving or copying from a TXTLIB). Any attempt to do so is ignored.

> **Note:** You may not <u>move</u> members from one TXTLIB to another TXTLIB if they are on the same minidisk. You may however, <u>copy</u> members between TXTLIBs on the same minidisk.



```
COPY --- FROM SPFDEMO.MYLIB.PLI  TO SPFDEMO.MASTER.PLI ----------------------
COMMAND ===>                                               SCROLL ===> PAGE
   NAME     RENAME    VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
S ACCOUNT              01.00  82/01/09   82/01/09 17:07    21    21     0  HUNTER
S ACCT1                01.01  82/02/11   82/04/23 14:52   199   193     0  BECKETT
S ACCT2                01.00  82/03/09   82/03/09 17:07    20    20     0  BECKETT
  COINS                01.04  82/04/24   82/04/28 16:20    19    19     4  JOSLIN
  COMPX                01.00  82/01/21   82/01/21 11:08    44    44     0  JOSLIN
S COMPY     ZCOMP_     01.01  82/01/14   82/01/16 12:30    13    13     1  JOSLIN
  DCLS                 01.00  82/04/23   82/04/23 15:14    20    20     0  LEESB
  LISTOUT              01.02  82/04/23   82/05/06 15:00    17    13     6  LEESB
  MAIN                 01.00  82/01/09   82/01/09 17:08     4     4     0  KIRK
  TESTDIR              01.02  82/04/23   82/05/06 09:04    30    43    10  LAIDLAW
  **END**
```

Figure 63 (Part 1 of 2). Move/Copy Utility - Copy Members from List



```
COPY --- FROM SPFDEMO.MYLIB.PLI  TO SPFDEMO.MASTER.PLI ----------------------
COMMAND ===> _                                                SCROLL ===> PAGE
   NAME     RENAME  VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
  ACCOUNT  *COPIED   01.00  82/01/09  82/01/09 17:07    21    21     0  HUNTER
  ACCT1    *COPIED   01.01  82/02/11  82/04/23 14:52   199   193     0  BECKETT
  ACCT2    *COPIED   01.00  82/03/09  82/03/09 17:07    20    20     0  BECKETT
  COINS              01.04  82/04/24  82/04/28 16:20    19    19     4  JOSLIN
  COMPX              01.00  82/01/21  82/01/21 11:08    44    44     0  JOSLIN
  COMPY    *COPIED   01.01  82/01/14  82/01/16 12:30    13    13     1  JOSLIN
  DCLS               01.00  82/04/23  82/04/23 15:14    20    20     0  LEESB
  LISTOUT            01.02  82/04/23  82/05/06 15:00    17    13     6  LEESB
  MAIN               01.00  82/01/09  82/01/09 17:08     4     4     0  KIRK
  TESTDIR            01.02  82/04/23  82/05/06 09:04    30    43    10  LAIDLAW
  **END**
```

Figure 63 (Part 2 of 2). Move/Copy Utility - Copy Members from List

PROJECT UTILITY (OPTION 3.4)

When you select this option, a panel is displayed that allows you to specify parameters for an ISPF library and an action to be performed (Figure 64). Possible actions are:

P   - Print project entries
blank - Display project entries

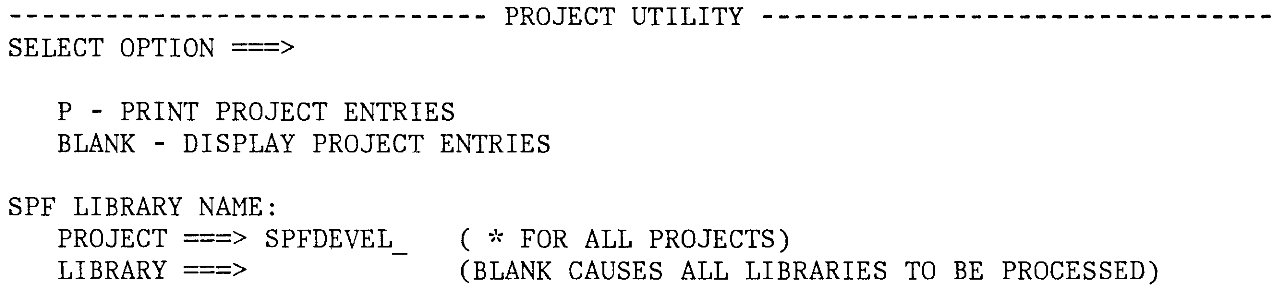
You may request information for:

- All libraries known to the system, by specifying an asterisk (*) for the project name.
- All libraries with a given project name, by specifying the project name and leaving the library name blank.
- All libraries with a given project and library name.

If you select option P (print project entries), a project listing is recorded in the ISPF list file. The listing contains CMS file identifications, locations, and file characteristics for each library with the specified project name.

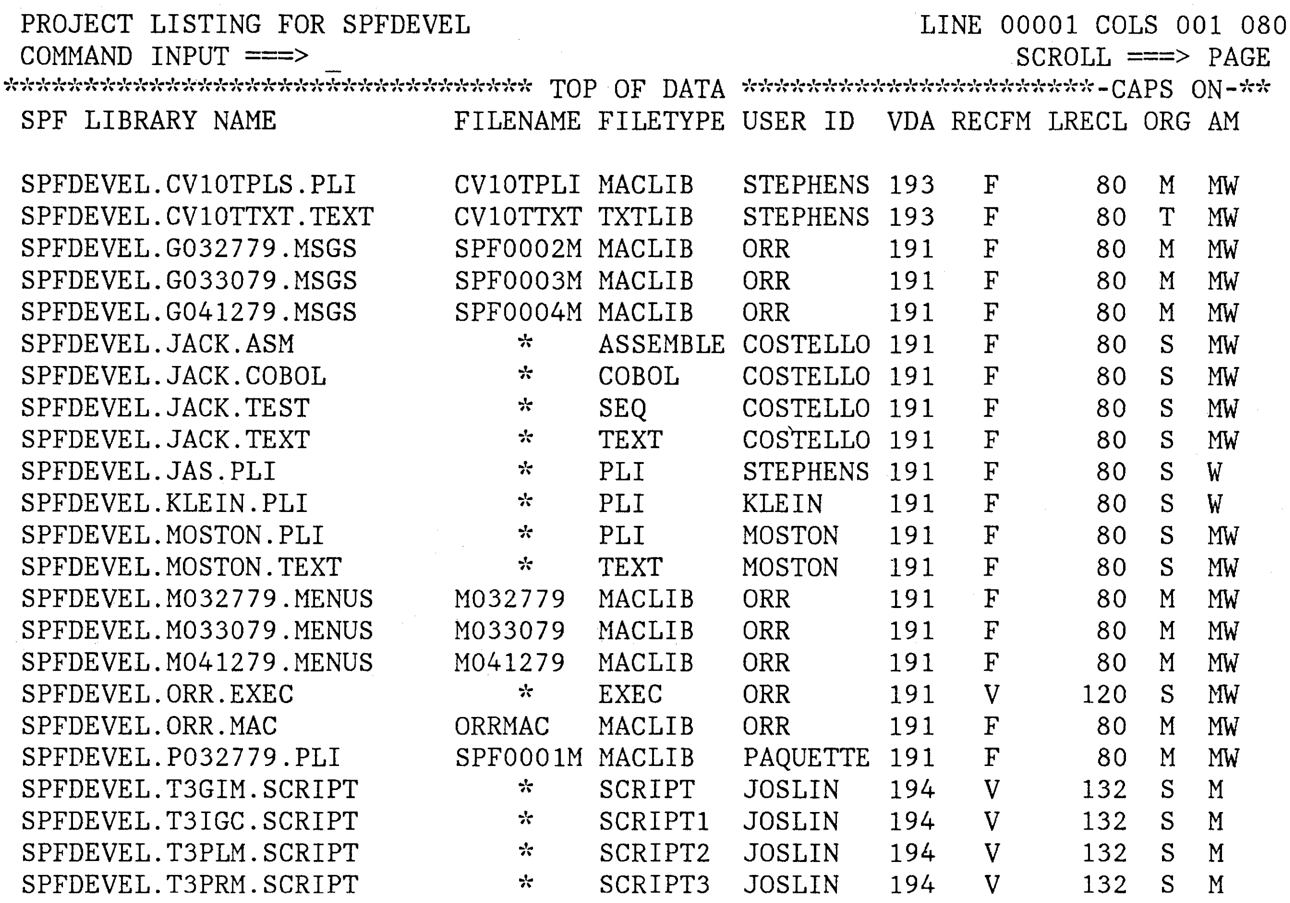
If you leave the option field blank (to display project entries), a project listing is displayed in browse mode. You can scroll the listing using the scrolling commands or the LOCATE command. You can use all browse commands. To terminate the display of the listing, enter the END command, which causes a return to the project utility panel.

Figure 65 shows an example of a 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 64. Project Utility Panel



```
 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 65. Project Utility - Sample Listing

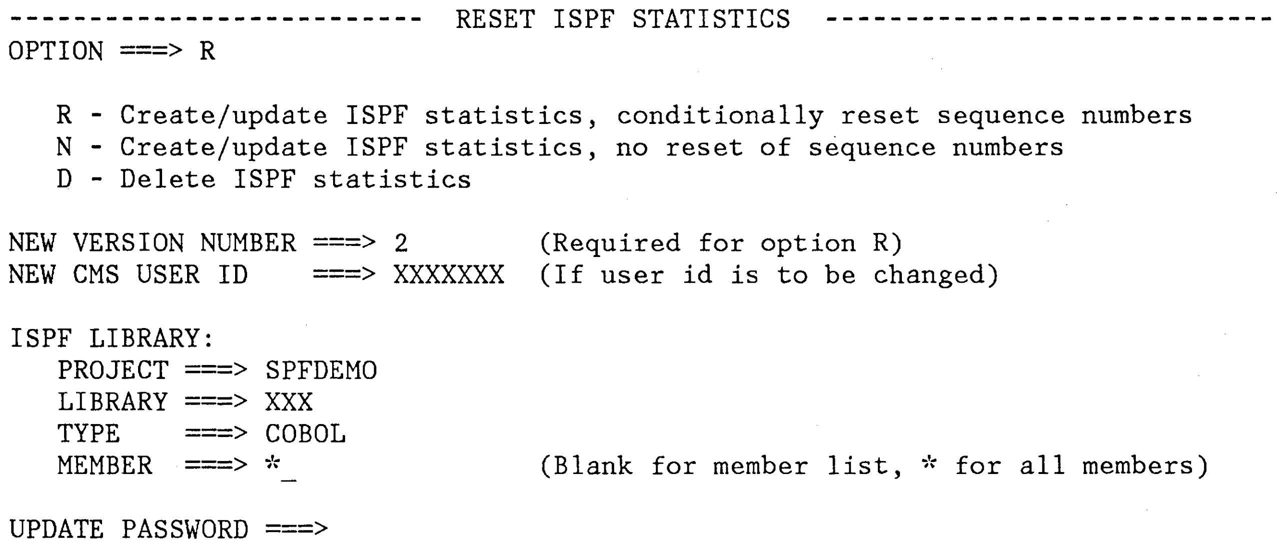
RESET STATISTICS UTILITY (OPTION 3.5)

When you select this option, a panel is displayed that allows you to create, update, or delete statistics and to reset sequence numbers (Figure 66). The reset utility handles only ISPF libraries. Refer to the section entitled "Library Statistics" in Chapter 3 for a discussion of the type of information maintained for each member of a library.

There are three valid options for this utility:

R - Create/update statistics, which conditionally resets sequence numbers
N - Create/update statistics, which does not reset sequence numbers
D - Delete statistics

Use the R option either to create statistics in a library that does not currently have them, or to update statistics in a library.



```
-------------------------- RESET ISPF STATISTICS ---------------------------
OPTION ===> R

   R - Create/update ISPF statistics, conditionally reset sequence numbers
   N - Create/update ISPF statistics, no reset of sequence numbers
   D - Delete ISPF statistics

NEW VERSION NUMBER ===> 2       (Required for option R)
NEW CMS USER ID    ===> XXXXXXX (If user id is to be changed)

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX
   TYPE    ===> COBOL
   MEMBER  ===> *_              (Blank for member list, * for all members)

UPDATE PASSWORD ===>
```

Figure 66. Reset Statistics Utility Panel

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.

Use the N option to create or update statistics as in option R; however, the data is not renumbered. Use this option (N) if the data already contains statistics and you want to update the user id and version information without renumbering the data.

Use the D option to delete statistics.

The version number field is required for option R, optional for option N, and ignored for option D. If you select option N and select a member without statistics, the statistics are created as though the version had been specified as "1".

If you specify a version number, 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 you specify a user id, it is placed in the user id field of the statistics. If you leave the user id field blank and select a member without statistics, the user id field of the statistics is 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, updated, or deleted 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, updated, or deleted for all members in the library.

If you request a member list, you may select members to be reset by entering an S in front of one or more member names. You may scroll the member list using the scrolling commands or the LOCATE command. To terminate the member list, enter the END command, which causes a return to the reset utility panel.

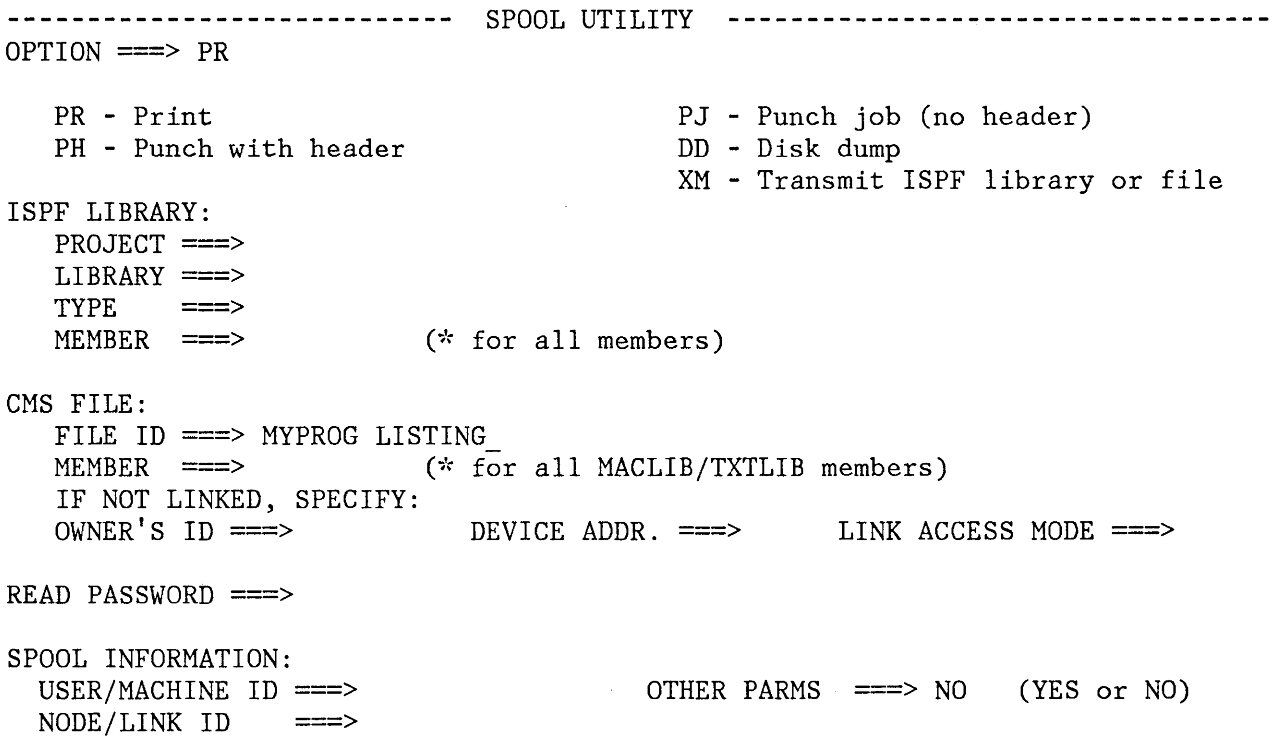
SPOOL UTILITY (OPTION 3.6)

When you select this option, a panel is displayed that allows you to specify a library or file and to indicate an action to be performed (Figure 67).

Possible actions are:

PR	- Print
PH	- Punch with header
PJ	- Punch job (no header)
DD	- Disk dump
XM	- Transmit

Note: You may not specify the ISPF list and log files; they may be printed only when you end your PDF session.



```
---------------------------  SPOOL UTILITY  --------------------------------
OPTION ===> PR

   PR - Print                                PJ - Punch job (no header)
   PH - Punch with header                    DD - Disk dump
                                             XM - Transmit ISPF library or file
ISPF 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:
  USER/MACHINE ID ===>                     OTHER PARMS  ===> NO   (YES or NO)
  NODE/LINK ID    ===>
```

Figure 67. Spool Utility - First Panel

Use option PR (print) to print a listing file.

Use the following options to send data to another CMS user:

- Option DD (disk dump) for any CMS file.
- Option PH (punch with header) for any CMS file that has a record length of 80 or less.
- Option XM (transmit) for any CMS file or an entire ISPF library (including library statistics).

Use option PJ (punch job) to send a file to another machine for job execution (you may use it for sending CMS jobs to a CMS batch machine, or JCL to a VS or VSE machine).

> **Note:** You may use the XM (transmit) option to send ISPF libraries or sequential files to an MVS node that has one of the TSO Extensions (TSO/E) Licensed Programs installed. Make sure that you specify the "new" transmission format (see below).

Enter an asterisk (*) in the member name field to process all members of a library, or a CMS MACLIB or TXTLIB.

The spool utility panel also allows you to specify:

- User/machine id - if the data is to be spooled to the virtual reader of another user or machine.

- Node/link 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.

When you have entered the desired information, a second panel is displayed if you specified "other parms - yes" on the first panel. Information entered on the second panel during previous uses of this utility is remembered and automatically redisplayed. This excludes "user/machine id" and "node/link id", which are simply copied from the first panel, but may be modified on the second panel.

For all options except XM (transmit), you may enter the following information on the second panel (Figure 68):

- Print command options - if option PR was specified. See the description of the PRINT command in VM/SP CMS Command and Macro Reference, SC19-6209.

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

- Spool class - for other than class A.

- Bin number - for other than your default destination.

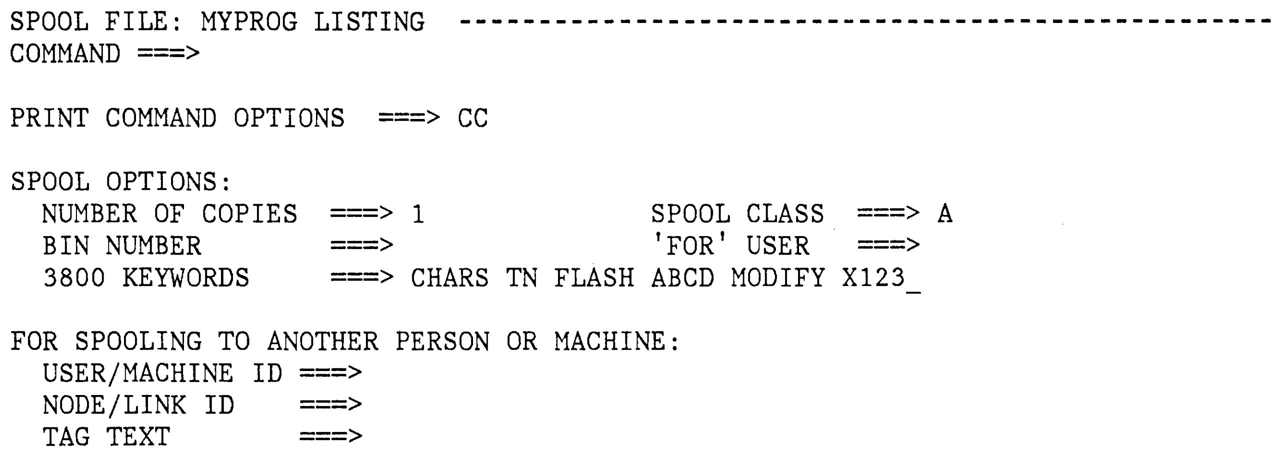
- 'For' user - to spool the output to another user's virtual printer on the same VM system (ignored if either "user/machine id" or "node/link id" is specified).

- 3800 keywords - if output is directed to an IBM 3800 printer that is attached to the CMS system as a virtual spooling device. See the description of the SPOOL command in VM/SP CP Command Reference for General Users, SC19-6211.

- User/machine id - to spool the output to the virtual reader of another user or machine.

- Node/link id - if the destination is a remote node on the network (valid with either "user/machine id" or "tag text", but not both).

- Tag text - to specify control parameters for printing on a remote non-VM system. See the description of the TAG command in VM/370 RSCS Networking Program Reference and Operations Manual, SH24-5005.



```
SPOOL FILE: MYPROG LISTING ---------------------------------------------------
COMMAND ===>

PRINT COMMAND OPTIONS  ===> CC

SPOOL OPTIONS:
  NUMBER OF COPIES  ===> 1             SPOOL CLASS  ===> A
  BIN NUMBER        ===>               'FOR' USER   ===>
  3800 KEYWORDS     ===> CHARS TN FLASH ABCD MODIFY X123_

FOR SPOOLING TO ANOTHER PERSON OR MACHINE:
  USER/MACHINE ID ===>
  NODE/LINK ID    ===>
  TAG TEXT        ===>
```

Figure 68. Spool Utility - Second Panel

For option XM (transmit), you may enter the following information on the second panel.

- Return acknowledgment (yes/no) - "yes" causes automatic generation of an acknowledgment message from the receiving user/machine. (The default is "yes" if the second panel is not displayed.)

- Transmission format (new/old) - specify "old" if you are transmitting to another node that has the previous System Productivity Facility for VM Program Product installed. In all other cases, specify "new".

- User/machine id - to spool the output to the virtual reader of another user or machine. (This parameter is required for option "XM".)

- Node/link id - if the destination user/machine is at a remote node on the network.

If you specified "other parms - no" on the first panel, the second panel is not displayed and the remembered information is not used.

PDF generates the appropriate command (e.g., PRINT, PUNCH, DISK DUMP), depending on the option selected, to spool the data to the specified destination. The first panel is then redisplayed to allow you to specify additional data to be spooled. When you enter the END command from the first panel, the spool file is closed (releasing the data for output) and you are returned to the next higher level panel.

Caution: Erroneous results may occur if you use this utility in split screen mode while you are also generating printer or punch output from the other logical screen. Output from the two logical screens may be intermingled on the virtual printer or punch if you jump back and forth between the screens.

READER UTILITY (OPTION 3.7)

The reader utility allows you to retrieve files from your virtual reader, including output (TEXT files, LISTING files, etc.) produced by the CMS batch machine as a result of PDF batch processing (option 5).

It also allows you to retrieve files that were sent to you from another user (on the same system or at a remote node). These files may include ISPF libraries or other partitioned or sequential data sets transmitted from an MVS node that has one of the TSO Extensions (TSO/E) Licensed Programs installed.

The options displayed by this utility are dependent upon the status and characteristics of the first active file in your virtual reader.

For a TEXT file produced by a PDF batch compilation or assembly (option 5) for which the source was an ISPF library, the options are:

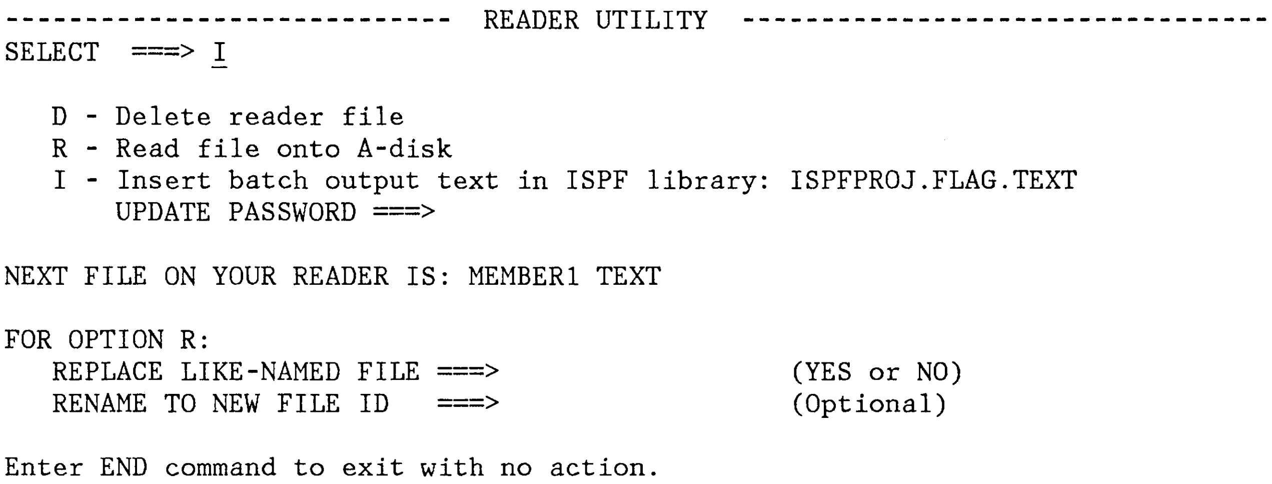
D - Delete reader file
R - Read file onto A-disk
I - Insert batch output TEXT in ISPF library

An example of this panel is shown in Figure 69. The default option in this case is I (pre-entered on the panel). If you use option I, you must specify an update password if the ISPF TEXT library (into which the file is to be inserted) is on a password-protected minidisk.

If you select option R, you may specify the following additional information:

- Replace like-named file (yes/no) - where "no" prevents a file with the same filename and filetype on your A-disk from being overwritten.

- Rename to new file id - which allows the file in the virtual reader to be renamed in the process of moving it to your A-disk.



```
---------------------------  READER UTILITY  --------------------------------
SELECT  ===> I

   D - Delete reader file
   R - Read file onto A-disk
   I - Insert batch output text in ISPF library: ISPFPROJ.FLAG.TEXT
       UPDATE PASSWORD ===>

NEXT FILE ON YOUR READER IS: MEMBER1 TEXT

FOR OPTION R:
   REPLACE LIKE-NAMED FILE ===>                    (YES or NO)
   RENAME TO NEW FILE ID   ===>                    (Optional)

Enter END command to exit with no action.
```

Figure 69. Reader Utility Panel

When you select an option and press ENTER, the first file in your virtual reader is processed. PDF then examines the next file in the reader and redisplays the reader utility panel with options that are appropriate for the file.

All variations of the reader utility panel include the D and R options. When you select the R option, ISPF issues a CMS command, such as READCARD or DISK LOAD, appropriate for the file being processed.

The other panels are similar to the one shown in Figure 69, except that:

1. If the first file in the reader is not a TEXT file for which the source was an ISPF library, the I option is not displayed.

2. If the first file in the reader is a transmitted file (sent via the XM option of the spool utility or transmitted from an MVS node using one of the TSO/E Licensed Programs), the following additional option is displayed:

   S - Store into ISPF library

The library identification fields (project, library, and type) are prefilled with the library identification as it appears in the reader. You may change this information to assign a new library identification. See Figure 70.

If you select option S and the library does not currently exist, a second panel is displayed to allow you to specify library attributes (Figure 70).

If you select option S and the library does exist, a confirmation panel is displayed with a warning that incoming members will replace like-named members in the existing library. As directed in the confirmation panel, press ENTER to continue or enter the END command to cancel.

3. If the first file in the reader is an acknowledgment from a transmit, the following additional option will be displayed:

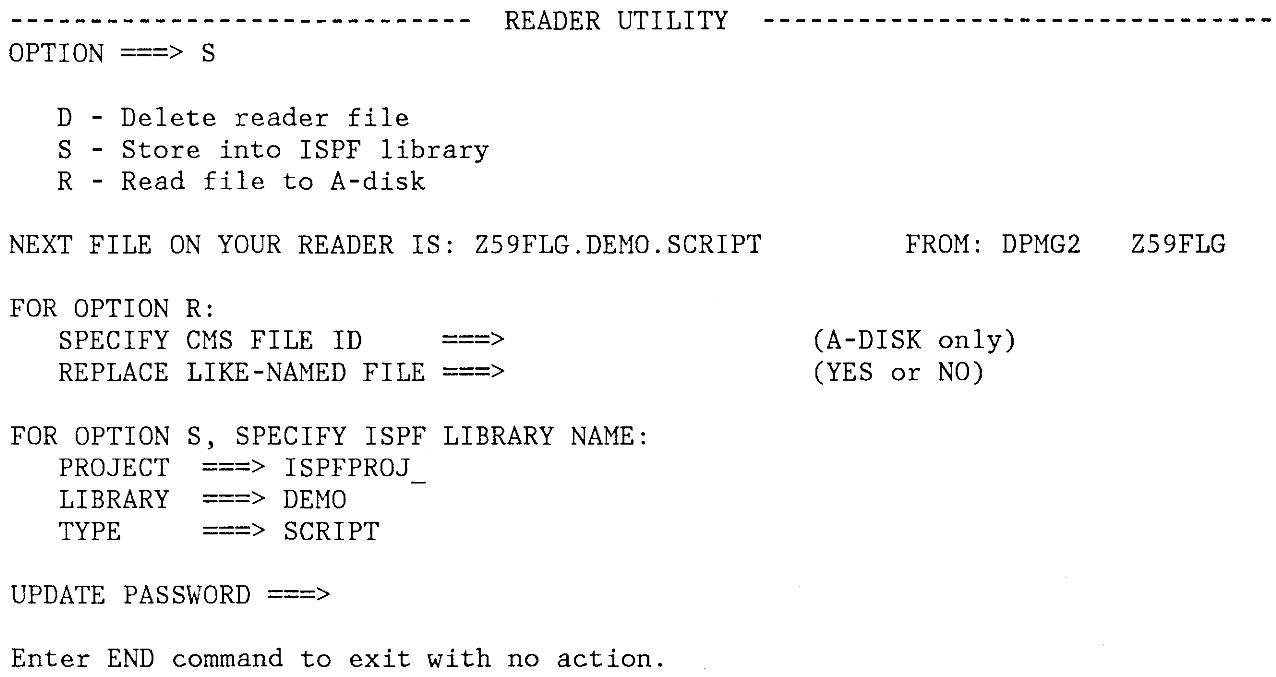
   L - Log message in MAIL LOG file

   This option causes the message to be added to the end of file 'MAIL LOG A' on your A-disk. If such a file does not currently exist, it is created.

4. If the reader is empty, or if the first file in the reader cannot be processed by this utility, one of the following messages is displayed on the second line of the panel:

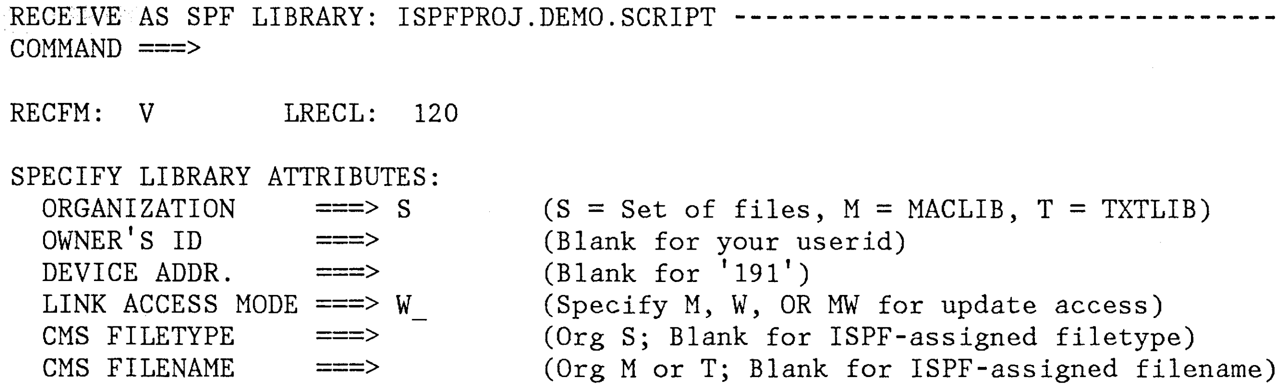
   ```
   NEXT FILE ON YOUR READER IS: READER IS EMPTY
   NEXT FILE ON YOUR READER IS: FILE IN HOLD STATUS
   ```

   In these cases, your only option is to return to the previous panel by either pressing the ENTER key or entering the END command. For files that are not class A or that are in hold status, you can enter a CMS command to change the class or status. Then reselect this utility to continue processing the contents of your virtual reader.



```
----------------------------  READER UTILITY  --------------------------------
OPTION ===> S

   D - Delete reader file
   S - Store into ISPF library
   R - Read file to A-disk

NEXT FILE ON YOUR READER IS: Z59FLG.DEMO.SCRIPT        FROM: DPMG2   Z59FLG

FOR OPTION R:
   SPECIFY CMS FILE ID     ===>                   (A-DISK only)
   REPLACE LIKE-NAMED FILE ===>                   (YES or NO)

FOR OPTION S, SPECIFY ISPF LIBRARY NAME:
   PROJECT  ===> ISPFPROJ_
   LIBRARY  ===> DEMO
   TYPE     ===> SCRIPT

UPDATE PASSWORD ===>

Enter END command to exit with no action.
```

Figure 70 (Part 1 of 2). Reader Utility - Receiving ISPF Libraries



```
RECEIVE AS SPF LIBRARY: ISPFPROJ.DEMO.SCRIPT --------------------------------
COMMAND ===>

RECFM:  V        LRECL:  120

SPECIFY LIBRARY ATTRIBUTES:
  ORGANIZATION     ===> S         (S = Set of files, M = MACLIB, T = TXTLIB)
  OWNER'S ID       ===>           (Blank for your userid)
  DEVICE ADDR.     ===>           (Blank for '191')
  LINK ACCESS MODE ===> W_        (Specify M, W, OR MW for update access)
  CMS FILETYPE     ===>           (Org S; Blank for ISPF-assigned filetype)
  CMS FILENAME     ===>           (Org M or T; Blank for ISPF-assigned filename)
```

Figure 70 (Part 2 of 2). Reader Utility - Receiving ISPF Libraries

RETRIEVE UTILITY (OPTION 3.8)

When you select this option, a panel (Figure 71) is displayed that allows you to specify a tape unit and serial number to retrieve ISPF libraries (via tape) from a TSO system. The tape drive should be attached to your machine and readied before you use this utility.

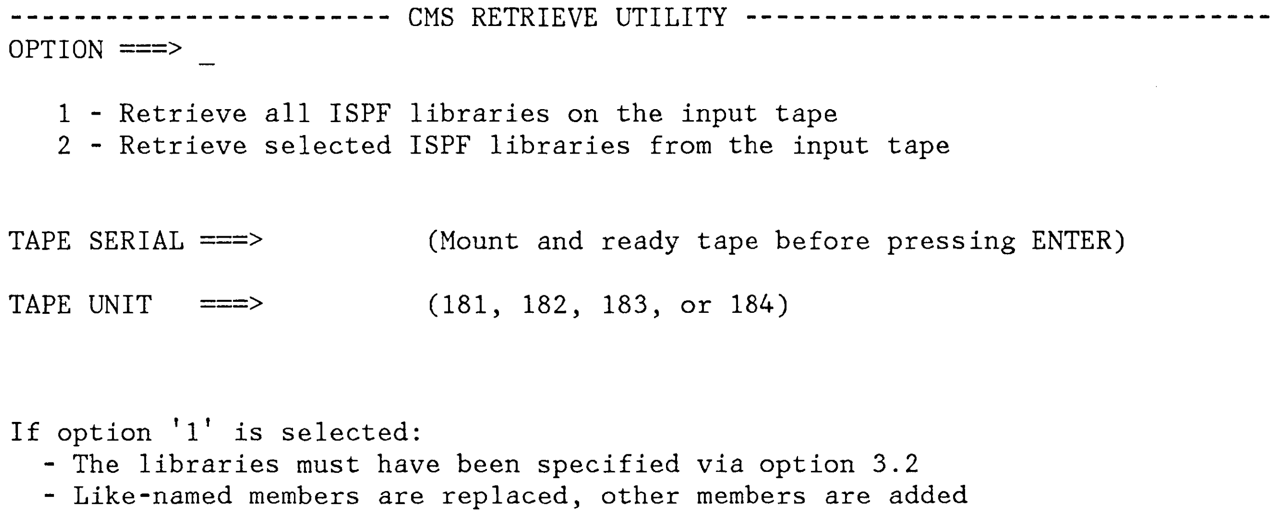
This utility has the following options:

1 - Retrieve all libraries on the input tape
2 - Retrieve selected libraries from the input tape

The tape must be a standard labeled tape created by the IEHMOVE utility. Any ISPF library (partitioned data set with a 3-level name) with fixed or variable record format can be processed. Library statistics are also retrieved during the processing.

> **Note:** This utility may not be used to place members into an ISPF TEXT library with organization T (TXTLIB).

When the end of tape is reached, or you enter the END or RETURN command, the tape is rewound but not unloaded.



```
----------------------- CMS RETRIEVE UTILITY -------------------------------
OPTION ===> _

   1 - Retrieve all ISPF libraries on the input tape
   2 - Retrieve selected ISPF 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 71. Retrieve Utility Panel

Retrieve All Libraries

When you select option 1, the libraries must have been previously specified to PDF with utility option 3.2 (specify library). The project name, library name, and type for each library on tape must exactly match an ISPF 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 be the same. 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

When you select option 2, the identification of each library encountered on tape may be reviewed at the terminal as the tape is read. You may then choose to retrieve the library, skip to the next library on the tape, or search the tape for a specified library. Optionally, you may rename a library as you retrieve it.

If you have not previously specified a library to PDF, you may specify its characteristics during the retrieval process. For libraries that you have already specified, the rules for record format and length are the same as for option 1, but the replacement of like-named members is under your control.

One of two secondary panels is displayed for each library read from tape (see Figure 72). The first panel shown in the figure is displayed if the library is already specified. You then have the following options:

M Merge members from tape with the specified ISPF library. For this option, you 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, you must enter the desired library identification (project name, library name, and type). If the designated library already exists, the first panel shown in Figure 72 is redisplayed. Normally, you would then select option M. If the designated library does not exist, the second panel shown in Figure 72 is displayed. Normally, you would then select option C and enter the library attributes.

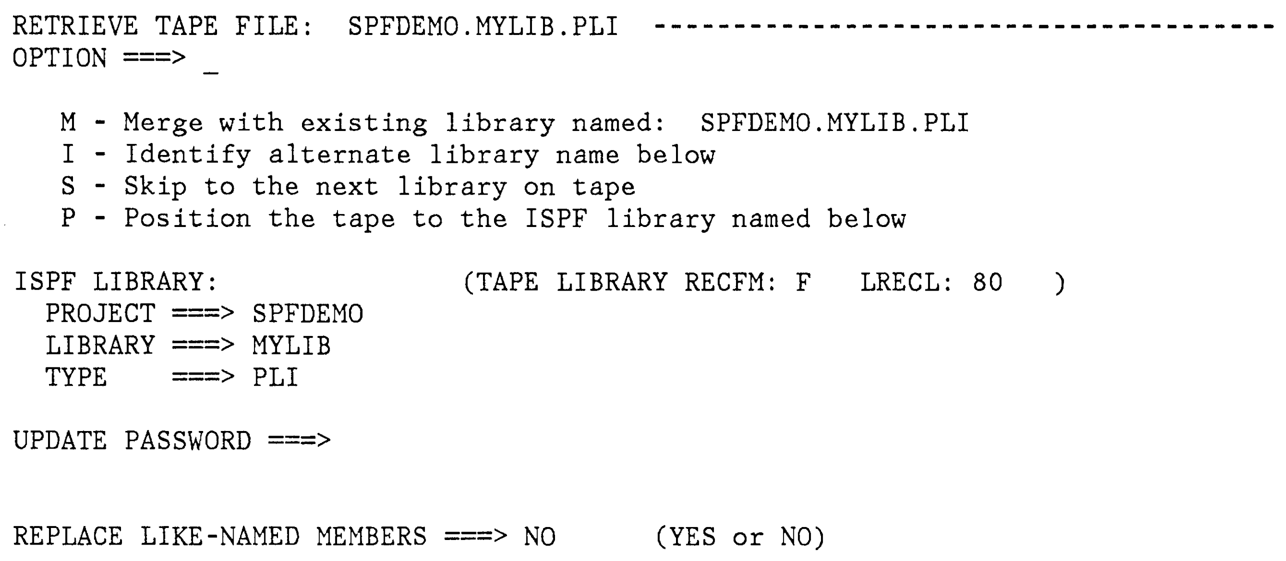
S Skip to the next library on the tape. A secondary panel is displayed for the next library, and you may choose the desired option.

P Position the tape to a specified library (and skip the current library). For this option, you must supply the project name, library name, and type for a library on the tape. If the library is found, a secondary panel is displayed, and you may choose the desired option. If the end of the tape is encountered before the library is found, the tape is rewound and the first retrieve panel is redisplayed. If the tape is positioned beyond the specified library, it is backspaced to that library.

If the library read from tape has not been specified, the second panel shown in Figure 72 is displayed. The first option on this panel is:

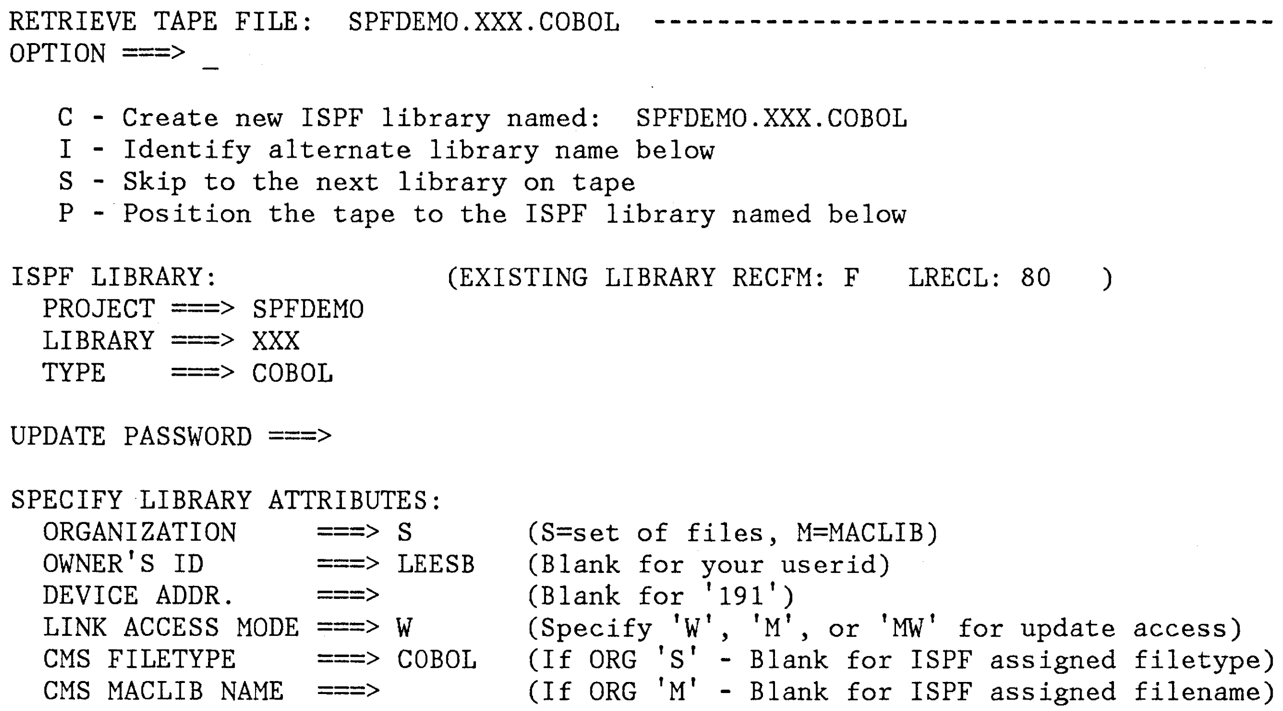
C Create new library. For this option, you must specify the library attributes on the lower part of the panel. This is the same information which may be specified under the file utility, option 3.2 (specify), except that the record format and length are taken from the tape library.

Other options on this panel are the same as those described above.



```
RETRIEVE TAPE FILE:  SPFDEMO.MYLIB.PLI  -------------------------------------
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 ISPF library named below

ISPF LIBRARY:                (TAPE LIBRARY RECFM: F   LRECL: 80   )
  PROJECT ===> SPFDEMO
  LIBRARY ===> MYLIB
  TYPE    ===> PLI

UPDATE PASSWORD ===>


REPLACE LIKE-NAMED MEMBERS ===> NO      (YES or NO)
```

Figure 72 (Part 1 of 2). Retrieve Utility - Secondary Panels



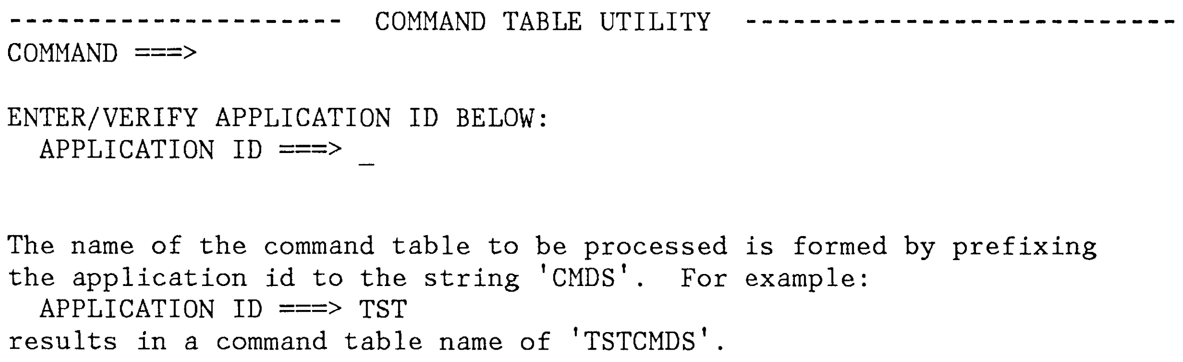
```
RETRIEVE TAPE FILE:  SPFDEMO.XXX.COBOL  -----------------------------------
OPTION ===> _

   C - Create new ISPF 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 ISPF library named below

ISPF 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       ===> LEESB   (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 ISPF assigned filetype)
  CMS MACLIB NAME  ===>         (If ORG 'M' - Blank for ISPF assigned filename)
```

Figure 72 (Part 2 of 2). Retrieve Utility - Secondary Panels

COMMAND TABLE UTILITY (OPTION 3.9)

The command table utility allows you to generate or modify application command tables. When you select this option, a panel (Figure 73) is displayed to prompt you for an application id. The name of the command table is then derived by appending "CMDS" to the application id. If the table exists in the table input library, it is displayed and may be modified. If the table does not exist in the table input library, a new table is generated.

> **Note:** You cannot use this utility to modify a command table that is currently in use. Command table ISPCMDS (the system command table) is always in use by the dialog manager. To modify the system command table, make a copy of the table, rename the copy, modify the copy, and then replace the original with the copy outside the ISPF environment.



```
--------------------  COMMAND TABLE UTILITY  -------------------------
COMMAND ===>

ENTER/VERIFY APPLICATION ID BELOW:
  APPLICATION ID ===> _


The name of the command table to be processed is formed by prefixing
the application id to the string 'CMDS'.  For example:
  APPLICATION ID ===> TST
results in a command table name of 'TSTCMDS'.
```

Figure 73. Command Table Utility Panel

The named table is displayed on an edit panel (Figure 74). Each row of the table shows the command verb, the truncation amount (T), and the action, with the description shown on a separate line, offset under the action.

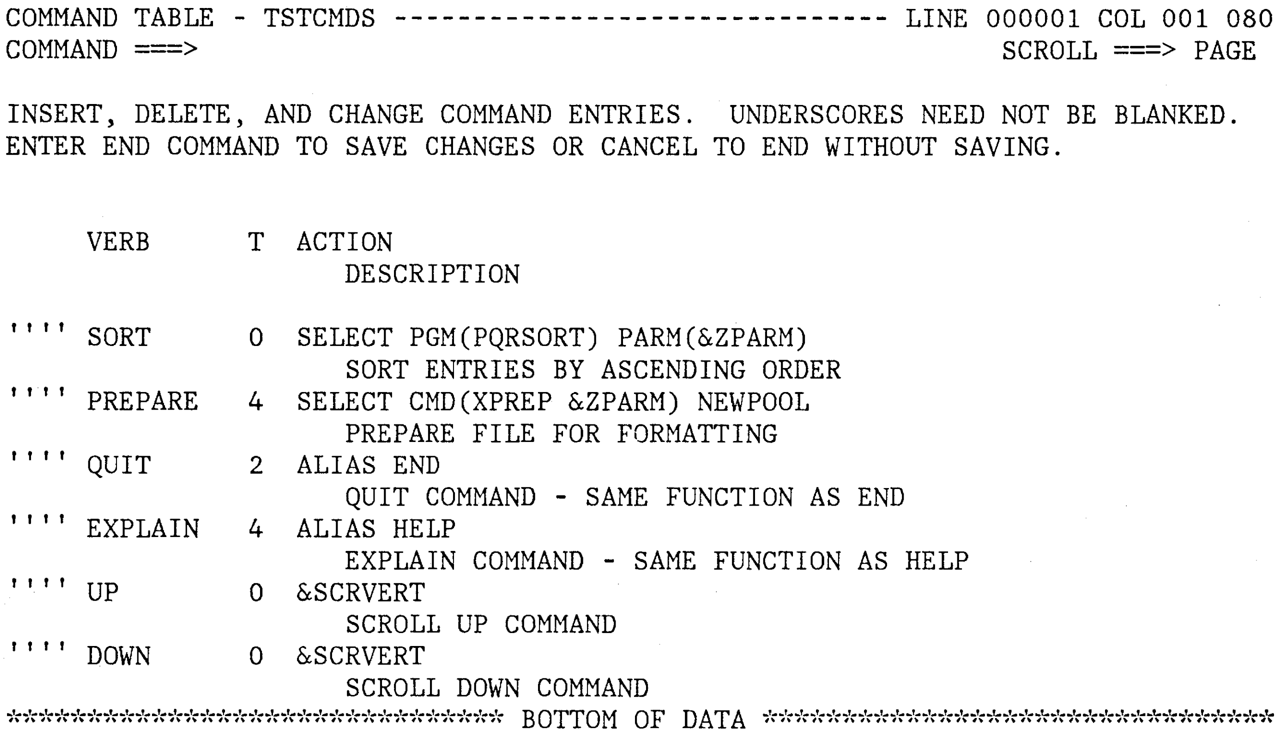
For a new table, this panel initially contains dummy entries in which all fields are underscores. The underscores are pad characters, and need not be blanked out.

You can scroll the table entries, and modify one or more entries simply by overtyping. In addition, you can enter the following line commands at the left of any entry by overtyping the four quote marks:

I or In Insert one or n lines. The inserted lines contain underscores (pad characters) in all field positions.

R or Rn Repeat one or n lines. The repeated lines contain underscores (pad characters) in the verb and truncation fields, but the action and description fields are copied from the line on which the R or Rn was entered.

D or Dn Delete one or n lines.



```
COMMAND TABLE - TSTCMDS ------------------------------ LINE 000001 COL 001 080
COMMAND ===>                                                 SCROLL ===> PAGE

INSERT, DELETE, AND CHANGE COMMAND ENTRIES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO SAVE CHANGES OR CANCEL TO END WITHOUT SAVING.


     VERB     T ACTION
                   DESCRIPTION

'''' SORT     0 SELECT PGM(PQRSORT) PARM(&ZPARM)
                   SORT ENTRIES BY ASCENDING ORDER
'''' PREPARE  4 SELECT CMD(XPREP &ZPARM) NEWPOOL
                   PREPARE FILE FOR FORMATTING
'''' QUIT     2 ALIAS END
                   QUIT COMMAND - SAME FUNCTION AS END
'''' EXPLAIN  4 ALIAS HELP
                   EXPLAIN COMMAND - SAME FUNCTION AS HELP
'''' UP       0 &SCRVERT
                   SCROLL UP COMMAND
'''' DOWN     0 &SCRVERT
                   SCROLL DOWN COMMAND
******************************* BOTTOM OF DATA ********************************
```

Figure 74. Command Table Editing Panel

In addition to the two scroll commands (UP and DOWN), you can enter the following primary commands:

END Causes the table to be saved in the table output library, and terminates the utility.

CANCEL Terminates without saving the table. CANCEL may be abbreviated to CAN.

You may enter multiple line commands or modifications in a single interaction. The lines are processed in the order in which they appear on the screen. Any line commands or modifications that are entered concurrently with the END command are processed before the table is saved.

Any null entries (in which at least the verb contains all underscores) are automatically deleted when the table is saved.

CONVERT MENUS/MESSAGES (OPTION 3.10)

This option is only for users who have installed the Structured Programming Facility (SPF) Program Product.

Installations that have previously extended or custom-tailored SPF must ensure that the primary option menu and all lower level selection panels (menus) that were displayed by the SPFUTIL program are in the new format. In ISPF, these menus are displayed by the SELECT service. The SPFUTIL program no longer exists.

This option automatically converts old-format (SPF) menus and messages to new-format ISPF panel and message definitions. When you select this option, two panels are displayed that are similar to the move/copy utility (option 3.3) panels. The first panel (Figure 75) allows you to specify whether you are converting menus or messages (option 1 or 2) and to specify the "from" library, which must be an ISPF library or MACLIB. The second panel (Figure 76) allows you to specify the "to" library, which must also be an ISPF library or MACLIB.

Old-format members are read from the first ("from") library, converted to the new format, and stored in the second ("to") library. The first library is not altered by this utility.

This utility is not capable of fully converting all old-format Structured Programming Facility (SPF) menus to ISPF panels. In general, this utility:

- Handles conversion of attribute characters, including those specified with the old-format <FIELDS> statement, provided the <FIELDS> statement is the first line encountered in the old-format menu.

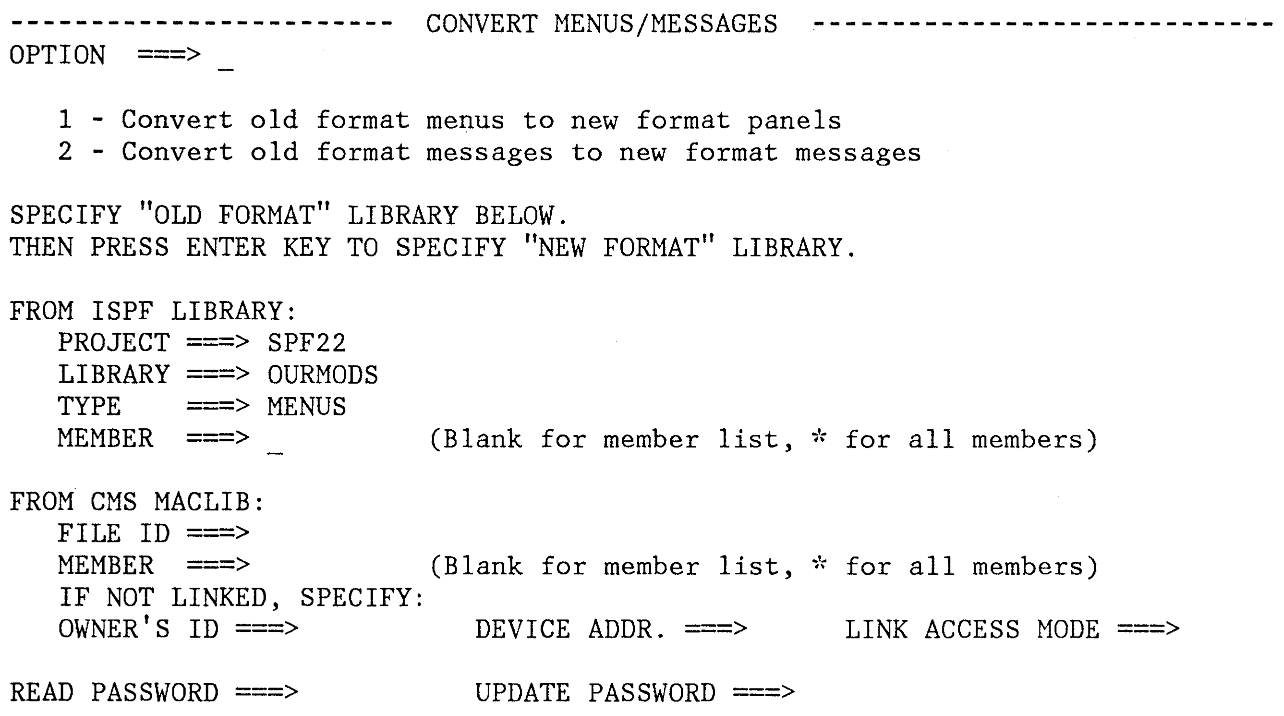
- Generates an)ATTR section, if required, and a)BODY section. The name of the first input field on line 2 (if any) is changed to ZCMD. Other input and output fields retain the same names as in the old-format menu.

- Generates an)INIT section with .HELP set to the name specified using the old GENHELP keyword, and with variables initialized as specified using the old INIT keywords.

The convert utility does not handle explicit cursor placement, the old KEY keyword, or the old LIST/RETURN keywords except as described below.

The convert utility fully converts:

- Old-format selection menus below the primary option level, that were designed to be processed by the SPFUTIL program in the Structured Programming Facility.



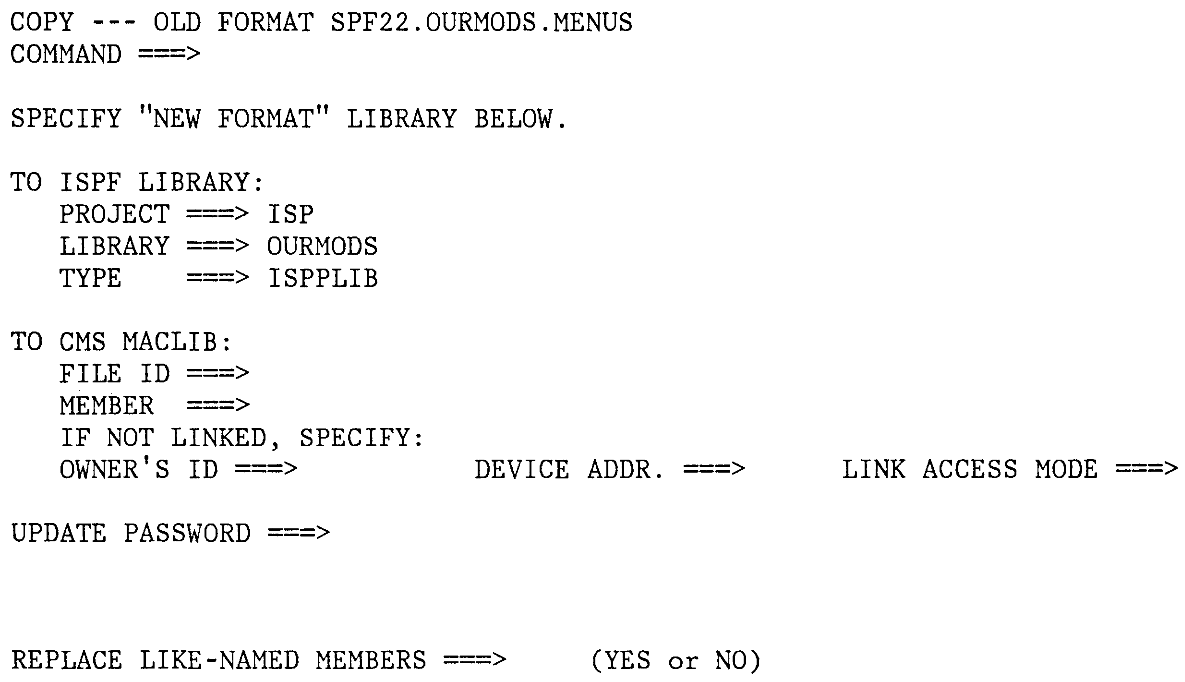
```
----------------------- CONVERT MENUS/MESSAGES ----------------------------
OPTION  ===> _

   1 - Convert old format menus to new format panels
   2 - Convert old format messages to new format messages

SPECIFY "OLD FORMAT" LIBRARY BELOW.
THEN PRESS ENTER KEY TO SPECIFY "NEW FORMAT" LIBRARY.

FROM ISPF LIBRARY:
   PROJECT ===> SPF22
   LIBRARY ===> OURMODS
   TYPE    ===> MENUS
   MEMBER  ===> _          (Blank for member list, * for all members)

FROM CMS MACLIB:
   FILE ID ===>
   MEMBER  ===>            (Blank for member list, * for all members)
   IF NOT LINKED, SPECIFY:
   OWNER'S ID ===>            DEVICE ADDR. ===>      LINK ACCESS MODE ===>

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

Figure 75. "From" Panel for Converting Menu/Message Definitions

- Old-format tutorial pages, except that the utility cannot handle the bypassing of a tutorial page that is displayed only if explicitly selected (bypassed in the normal flow when you keep pressing the ENTER key). The converted page is not bypassed in the normal flow. To correct the problem, you must manually change the (new-format) parent panel by inserting an asterisk in front of the panel name in the TRANS statement. For more information, see "Help/Tutorial Panels" in ISPF Dialog Management Services.

For messages, the utility properly handles the text of both short and long messages, the HELP reference (if any), and the ALARM indicator. Variable fields in old-format messages are converted to dummy variable names, beginning with an ampersand. You must change these manually to the appropriate dialog variable names.

Only new-format messages may be displayed on new-format panels, and only old-format messages may be displayed on old-format panels. The ISPF LOG service writes only new-format messages to the ISPF log file.



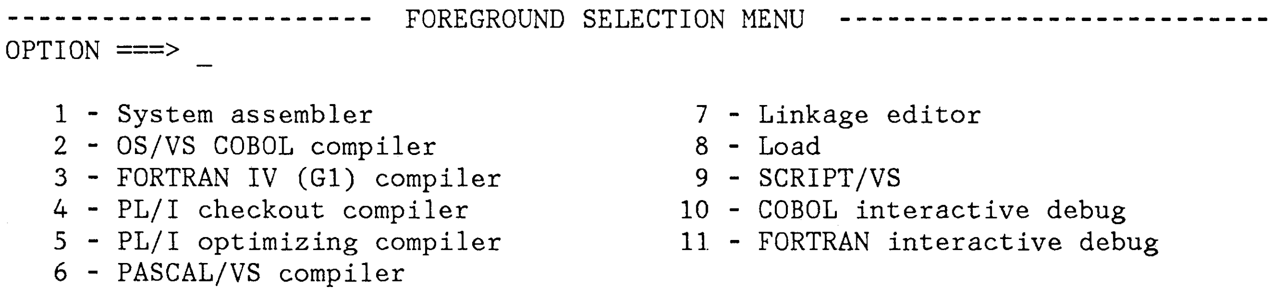
```
COPY --- OLD FORMAT SPF22.OURMODS.MENUS
COMMAND ===>

SPECIFY "NEW FORMAT" LIBRARY BELOW.

TO ISPF LIBRARY:
   PROJECT ===> ISP
   LIBRARY ===> OURMODS
   TYPE    ===> ISPPLIB

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

UPDATE PASSWORD ===>



REPLACE LIKE-NAMED MEMBERS ===>     (YES or NO)
```

Figure 76. "To" Panel for Converting Menu/Message Definitions

CHAPTER 8. FOREGROUND (OPTION 4)

The foreground option provides an interface to standard language processors for foreground compilation and assembly of programs stored in libraries or CMS files. It also provides an interface to the Document Composition Facility (SCRIPT/VS) Program Product for formatting, browsing, and printing documents, to the FORTRAN and COBOL interactive debug facilities, to the linkage editor, and to the CMS LOAD command.

The foreground selection panel is shown in Figure 77.



```
----------------------  FOREGROUND SELECTION MENU  ---------------------------
OPTION ===> _

   1 - System assembler                     7 - Linkage editor
   2 - OS/VS COBOL compiler                 8 - Load
   3 - FORTRAN IV (G1) compiler             9 - SCRIPT/VS
   4 - PL/I checkout compiler              10 - COBOL interactive debug
   5 - PL/I optimizing compiler            11 - FORTRAN interactive debug
   6 - PASCAL/VS compiler
```

Figure 77. Foreground Selection Panel

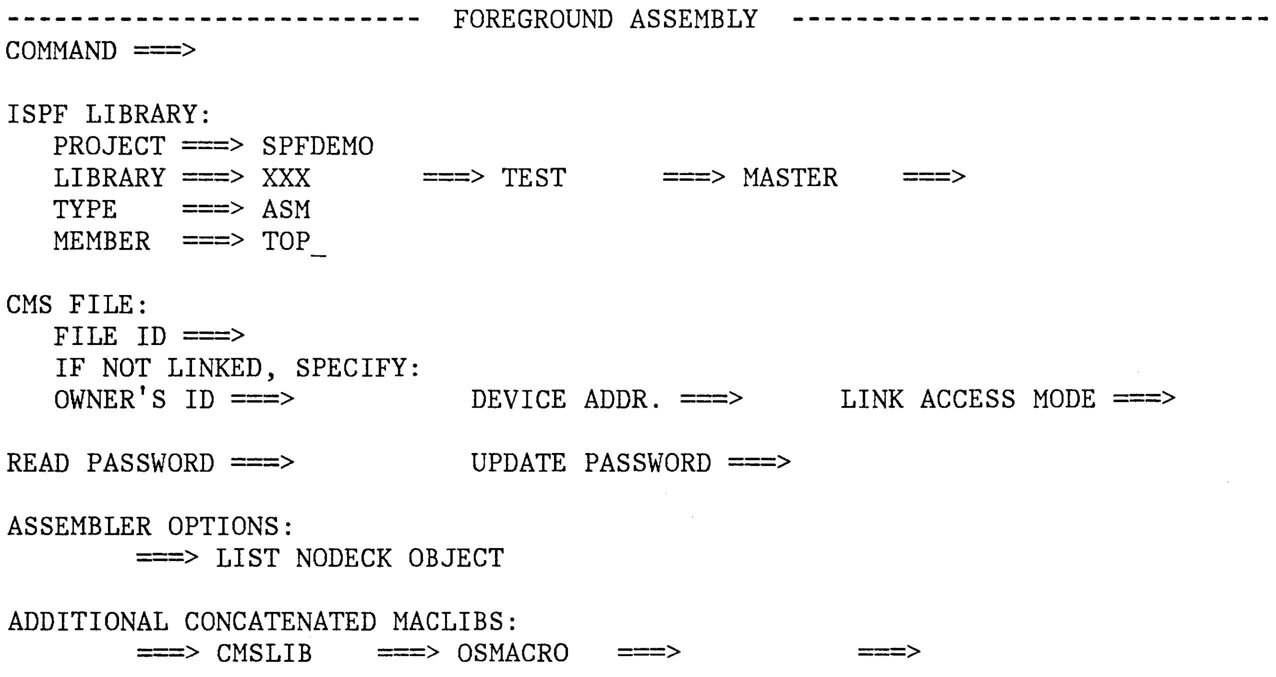
PROCESSING SEQUENCE

The sequence for foreground processing is:

1. Select an option from the foreground selection panel and press ENTER.

2. Specify library information or a CMS fileid and processor options on the entry panel displayed for the selected foreground processor. Figure 78 shows an example for the assembler. The project name, first library name, and type qualifier must be specified to assemble a program whose source is in a library. The fileid must be specified to assemble a CMS file. It is acceptable to specify only the filename; the filetype defaults to ASSEMBLE. (If the CMS filename is entered, it overrides any library specification.) The other fields on the panel are optional. See "Assembler/Compiler Processing" for a description of these fields.

3. If the input is from an ISPF library and you do not specify a member name, a member list is displayed. You may select a member from the list by entering an S in front of the member name.



```
-------------------------- FOREGROUND ASSEMBLY ------------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX       ===> TEST       ===> MASTER     ===>
   TYPE    ===> ASM
   MEMBER  ===> TOP_

CMS FILE:
   FILE ID ===>
   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 78. Foreground - Assembler Entry Panel

4. For SCRIPT/VS processing, a second entry panel is displayed, allowing you to specify formatting options for draft or final document processing. See "SCRIPT/VS Processing."

5. Once all of the input parameters have been specified, the foreground processor is invoked.

6. The GLOBAL command and the command that invokes the language processor is displayed along with 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. Upon completion of the foreground processing, a "MORE..." message is displayed in the lower right corner of the screen. Press the PA2 key. If the foreground processor generated an output listing, the listing file is displayed automatically in browse mode. The output can be scrolled up or down using the scrolling commands. All the browse commands are available to you. When you have finished browsing the listing, enter the END command.

7. Following the browsing of the output listing, the foreground print panel is displayed to allow you to print, keep, or delete the output (Figure 79). On this panel, the "file id" field shows the id of the output listing file. When you press the ENTER key, the processor entry panel is redisplayed.

   If you select option PK (print file and keep) or PD (print file and delete), you may specify the following spool parameters:

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

   - Spool class - for other than class A.

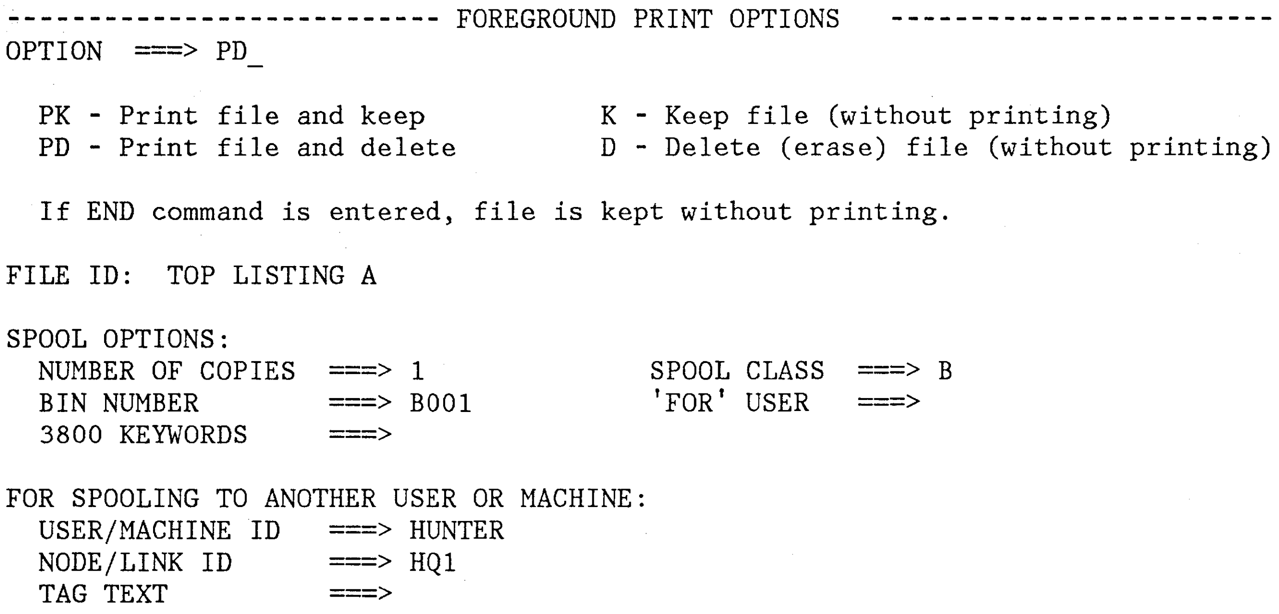
   - Bin number - for other than your default destination.

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

   - 3800 keywords - if output is directed to an IBM 3800 printer that is attached to the CMS system as a virtual spooling device. See the description of the SPOOL command in VM/SP CP Command Reference for General Users, SC19-6211.

   - User/machine id - to spool the output to the virtual reader of another user or machine.

   - Node/link id - if the destination is a remote node on the network (valid with either "user/machine id" or "tag text", but not both).



```
-------------------------- FOREGROUND PRINT OPTIONS  -----------------------
OPTION  ===> PD_

  PK - Print file and keep          K - Keep file (without printing)
  PD - Print file and delete        D - Delete (erase) file (without printing)

  If END command is entered, file is kept without printing.

FILE ID:  TOP LISTING A

SPOOL OPTIONS:
  NUMBER OF COPIES  ===> 1             SPOOL CLASS  ===> B
  BIN NUMBER        ===> B001          'FOR' USER   ===>
  3800 KEYWORDS     ===>

FOR SPOOLING TO ANOTHER USER OR MACHINE:
  USER/MACHINE ID   ===> HUNTER
  NODE/LINK ID      ===> HQ1
  TAG TEXT          ===>
```

Figure 79. Foreground Print Panel

- Tag text - to specify control parameters for printing on a remote non-VM system. See the description of the TAG command in VM/370 RSCS Networking Program Reference and Operations Manual, SH24-5005.

8. When the entry panel is redisplayed, a message indicating completion of the process is displayed in the upper right corner of the screen. Enter other parameters and invoke the processor again, or enter the END command to return to the foreground selection panel.

Note: If a foreground option terminates abnormally, a termination message is displayed in the upper right corner of the screen and browse mode is not entered. The output listing is retained, but the foreground print panel is not displayed.

ASSEMBLER/COMPILER PROCESSING

The assembler entry panel shown in Figure 78 is typical of the language processor panels. In the figure, a library concatenation sequence of three ISPF libraries has been specified. The concatenation order is:

```
SPFDEMO.XXX.ASM
SPFDEMO.TEST.ASM
SPFDEMO.MASTER.ASM
```

Note: The type qualifier for an assembler source library need not be ASSEMBLE, as illustrated in the example.

Before invoking the foreground processor, PDF scans 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) is created on your 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 your 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 PDF are erased at the completion of the foreground process.

For this example, if SPFDEMO.XXX.ASM is a collection 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 ISPF-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 PDF 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 libraries, by specifying them in the desired order in the "additional concatenated MACLIBs" fields.

3. If the source is from an ISPF library, the TEXT file is placed in an ISPF library with a three-level name composed of the project name, the first library name, and a type qualifier of TEXT. The member name for the TEXT file is the same as the source member. The

LISTING file is placed on your 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 ISPF 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 your A-disk.

5. In specifying a CMS file that would require a LINK and ACCESS, you 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 you specify them. Only those options you specify on the panel are passed to the language processor; ISPF does not automatically generate any options.

The entry panels for the various 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 ISPF library. Rather, it is left on your 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 your 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 your A-disk.

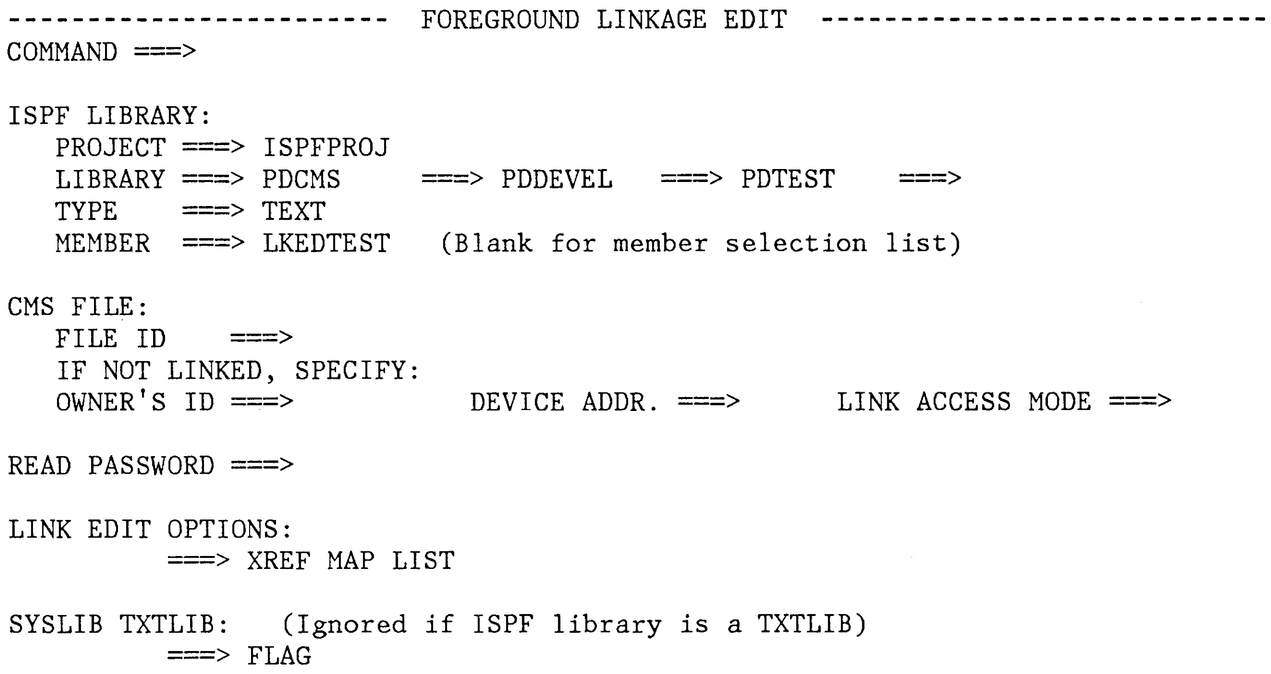
The entry panels for the interactive debug facilities differ somewhat from the other foreground panels:

- The COBOL interactive debug panel allows you to specify up to four additional program ids (which must match the external filenames), and PRINT and SOURCE options which determine what FILEDEFs are generated by ISPF. You may also specify up to four additional TXTLIBs.

- The FORTRAN interactive debug panel 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.

LINKAGE EDITOR PROCESSING

For the linkage editor, the CMS LKED command is invoked from the entry panel (Figure 80). ISPF generates the command from the input parameters you specify and then issues the command.



```
------------------------  FOREGROUND LINKAGE EDIT  ----------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> ISPFPROJ
   LIBRARY ===> PDCMS     ===> PDDEVEL    ===> PDTEST     ===>
   TYPE    ===> TEXT
   MEMBER  ===> LKEDTEST   (Blank for member selection list)

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

READ PASSWORD ===>

LINK EDIT OPTIONS:
          ===> XREF MAP LIST

SYSLIB TXTLIB:   (Ignored if ISPF library is a TXTLIB)
          ===> FLAG
```

Figure 80. Foreground - Linkage Editor Example

The linkage editor input is contained in a single member of one of the specified ISPF libraries or in a CMS file with a filetype of "TEXT". Specifying a CMS file overrides the specification of a member of an ISPF library.

The input text libraries may be password protected; specify the password on the entry panel. It is not displayed, but is retained by PDF. All input libraries must have the same read password.

Additional text information may be included in a link edit if it is available in other text files or in a TXTLIB. Text information may reside in up to four ISPF libraries and in a TXTLIB that you link and access and then specify on the entry panel.

> **Note:** If the you leave the "file-id" field blank and include an ISPF library that is defined as a TXTLIB, any TXTLIB that you specify in the "SYSLIB TXTLIB" field will be ignored. Since only one TXTLIB can be specified as SYSLIB input to the LKED command, you should ensure that any ISPF-defined input library is a set of sequential files, rather than a TXTLIB.

PDF does not automatically specify any linkage edit options. The options you specify on the panel are passed directly to the LKED command.

The LKED command produces two output files on your A-disk. It produces a file containing the load module that the linkage editor created, with the member name of the input file as the file name and a file type of LOADLIB. It also produces a listing file containing output according to the XREF, MAP, or LIST options you select. The file name of the listing file is the member name of the input file; the file type is LKEDIT. When the linkage edit completes, the listing is automatically displayed in browse mode, unless a termination or unrecoverable error occurs. If you specify the linkage editor option NOPRINT, no listing file is produced. If you specify PRINT, the listing file is sent directly to the printer and will no longer exist. To retain the listing for browsing, do not specify either of these options.

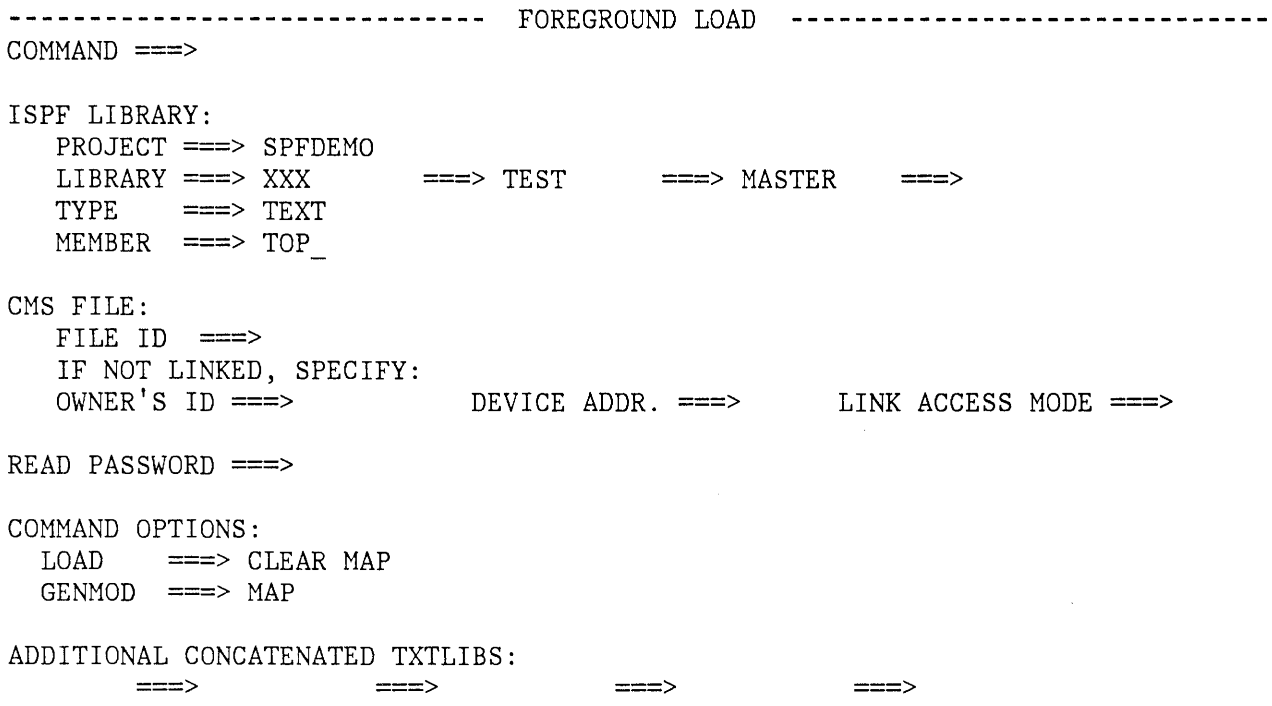
If you make entries in fields like those on the entry panel in Figure 80, some of the commands invoked might look like:

```
FILEDEF SYSLIB DISK PDCMS TXTLIB * (PERM
GLOBAL TXTLIB PDCMS PDDEVEL PDTEST FLAG
LKED LKEDTEST (XREF MAP LIST )
```

assuming that all the libraries are TXTLIBs. Note that only the first ISPF library is used as SYSLIB input.

LOAD PROCESSING

This foreground option (option 4.8) provides an interface to the LOAD command (Figure 81). The entry panel allows a simple LOAD and GENMOD command to be issued with the specified options.



```
------------------------------  FOREGROUND LOAD  ------------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX        ===> TEST       ===> MASTER     ===>
   TYPE    ===> TEXT
   MEMBER  ===> TOP_

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

READ PASSWORD ===>

COMMAND OPTIONS:
  LOAD    ===> CLEAR MAP
  GENMOD  ===> MAP

ADDITIONAL CONCATENATED TXTLIBS:
        ===>           ===>           ===>           ===>
```

Figure 81. Foreground - LOAD Panel

SCRIPT/VS PROCESSING

When this option is selected, a sequence of panels is displayed to allow you to format, display, and print text maintained in ISPF libraries or sequential CMS files. Use of this facility requires the installation of the Document Composition Facility (SCRIPT/VS) Program Product, with the Foreground Environment Feature.

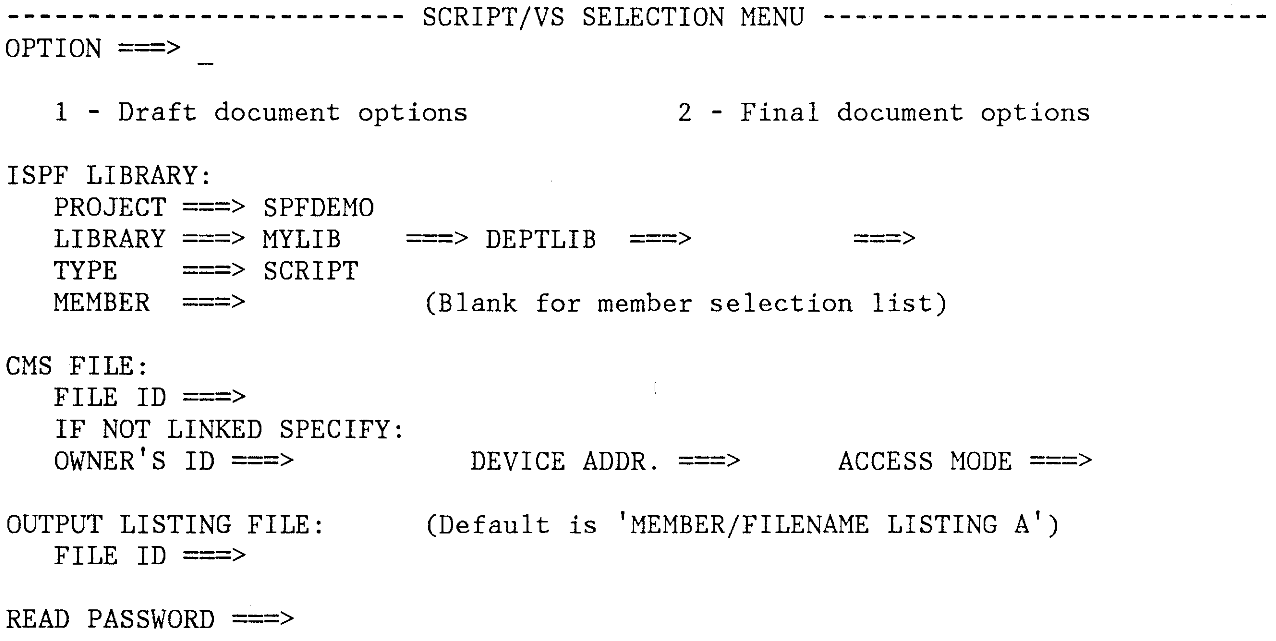
This facility has the following options:

  1 - Draft document options
  2 - Final document options

For the draft document options, it is assumed that your primary interest is to assure that the content, syntax, and spelling within the document are correct. The final document options provide additional flexibility and control for specifying document formats.

The first panel displayed by the SCRIPT/VS option allows you to select the draft or final options, and to specify the library or file to be formatted (Figure 82). For ISPF libraries, a member list is displayed if a member name is not specified. You may specify a concatenated sequence of ISPF libraries to allow inclusion of imbedded or appended members from more than one library.

Any ISPF libraries to be processed must have an organization of S (set of sequential files). They need not have an ISPF type qualifier of SCRIPT, and the corresponding set of files need not have a CMS filetype of SCRIPT. If, however, you specify a CMS file id (on the second part of the panel), any imbedded or appended files must have a filetype of SCRIPT unless they are explicitly defined with a SCRIPT control word.

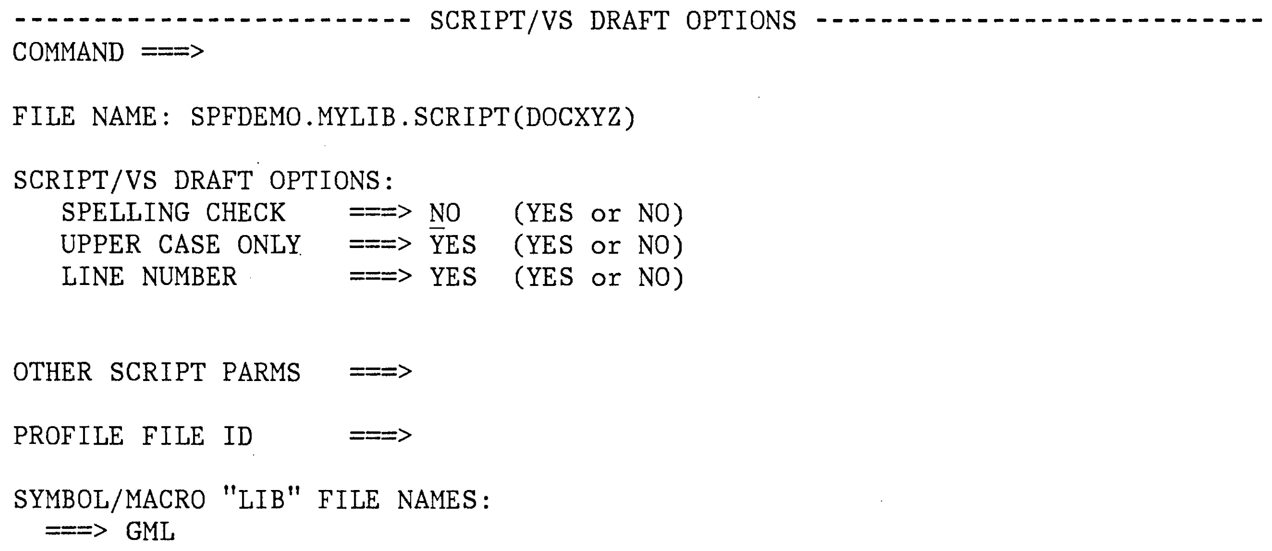


```
------------------------- SCRIPT/VS SELECTION MENU ---------------------------
OPTION ===> _

   1 - Draft document options             2 - Final document options

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> MYLIB    ===> DEPTLIB  ===>          ===>
   TYPE    ===> SCRIPT
   MEMBER  ===>            (Blank for member selection list)

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

OUTPUT LISTING FILE:       (Default is 'MEMBER/FILENAME LISTING A')
   FILE ID ===>

READ PASSWORD ===>
```

Figure 82. SCRIPT/VS Utility - First Panel

Draft Document Options

When you select option 1 on the first SCRIPT/VS panel, a secondary panel (Figure 83) is displayed to allow you to specify the draft document options. All the parameters on this panel are optional.



```
------------------------ SCRIPT/VS DRAFT OPTIONS ---------------------------
COMMAND ===>

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 83. SCRIPT/VS Draft Document Options

For automatic spelling verification, enter "yes" in the spelling check field. You must also enable spelling verification within the document. (See the .sv control word in the Document Composition Facility User's Guide, SH20-9161.)

For all-uppercase printing, enter "yes" in the upper case only field.

For line numbers to be printed in the output document, enter "yes" in the line number field. Line numbers can be useful for correcting errors and misspelled words, which SCRIPT/VS identifies by line number.

Use the "other SCRIPT parms" field to enter any SCRIPT/VS parameters not on the panel.

If a SCRIPT/VS document profile is to be used, you must enter the name of the CMS file that contains the profile in the "profile" field. Enter a file id of PROFILE SCRIPT for the default profile.

To incorporate one or more user generalized markup language (GML) libraries, 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 one is defined, is used.

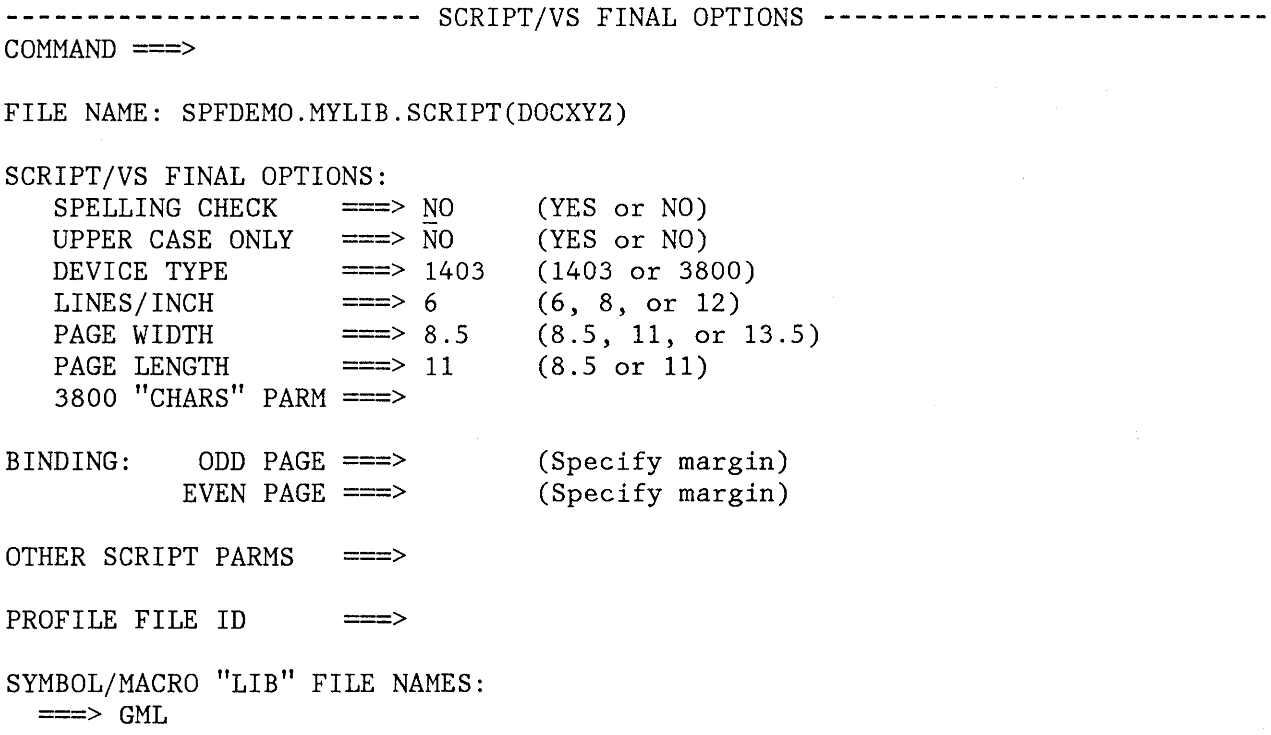
Final Document Options

When you select option 2 on the first SCRIPT/VS panel, a secondary panel (Figure 84) is displayed to allow you to specify the final document options. All the parameters on this panel are optional. However, if you enter a device type, you must fill in the lines per inch and form definition fields correctly.

For automatic spelling verification, enter "yes" in the spelling check field. You must also enable spelling verification within the document. (See the .sv control word in the Document Composition Facility User's Guide, SH20-9161.)

For all-uppercase printing, enter "yes" in the upper case only field.

The next four fields on the panel are used to describe the output device and the physical characteristics of the printed page. Specify the type of printer on which the output will be printed (and for which SCRIPT/VS will format the document) in the device type field.



```
-------------------------- SCRIPT/VS FINAL OPTIONS ----------------------------
COMMAND ===>

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 84. SCRIPT/VS Final Document Options

If the document is to be printed on a device other than an IBM 1403 or 3800, leave this field blank, and specify the printer 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 fit on the printed page.

The 3800 chars parm field allows you to specify the names of one or more type fonts if the document is being formatted for an IBM 3800 printer. You may enter the names of one or two SCRIPT/VS fonts or of up to four 3800 character arrangement tables, separated by blanks. You must specify these same parameters on the foreground print panel with multiple CHARS parameters and an OPTCD=J parameter.

You may specify the page margin to be used in binding the document by entering the size of the margin in the binding fields. You can specify a different margin for even and odd numbered pages. For more information, see "BIND" in Document Composition Facility User's Guide, SH20-9161.

Use the "other SCRIPT parms" to enter any SCRIPT/VS parameters not on the panel.

If you use the "other SCRIPT parms" field to specify a device other than the 1403 or 3800, you must have previously added the device to the SCRIPT/VS device characteristics table. You may select a device from this table by leaving the device type field blank and specifying the SCRIPT/VS device parameter in the other SCRIPT parms field.

If a SCRIPT/VS document profile is to be used, you must enter the name of the CMS file that contains the profile in the "profile" field. Enter a file id of PROFILE SCRIPT for the default profile.

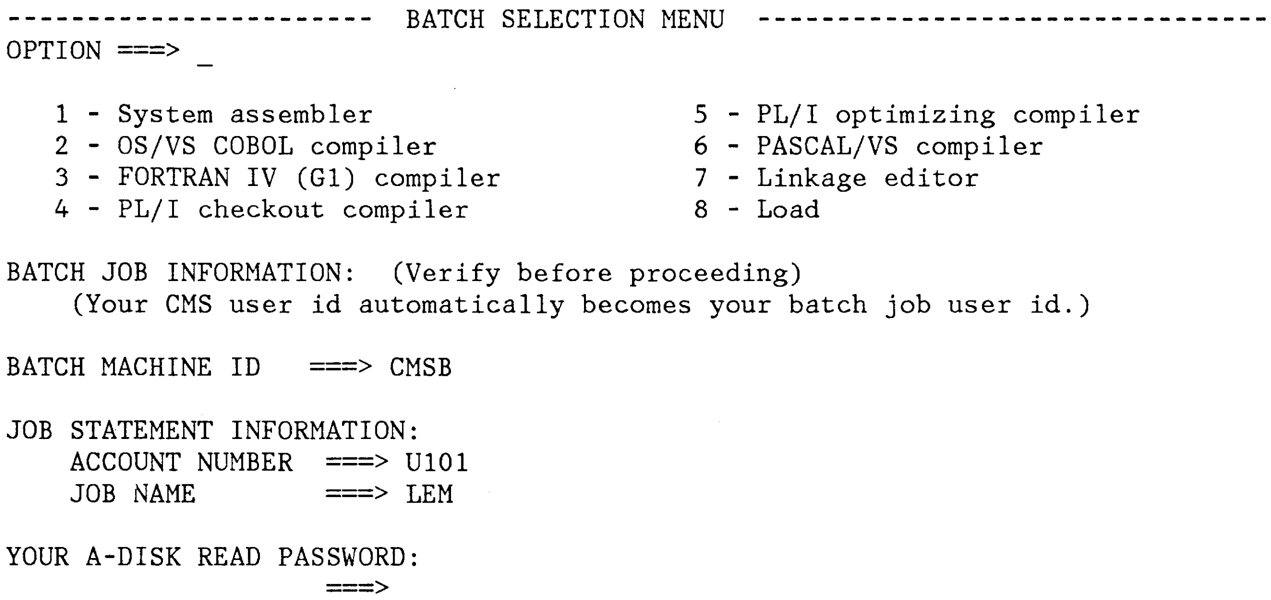
To incorporate one or more user GML libraries, enter the filename(s) in the "symbol/macro lib" field. These files must be MACLIBs. If no files are specified, the default GML library, if one is defined, is used.

CHAPTER 9. BATCH (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 ISPF libraries or CMS files. It also provides an interface for batch execution of the CMS LKED (linkage edit) and LOAD commands.

The batch selection panel is shown in Figure 85. It allows you to enter job statement parameters and to select a language processor. If your A-disk is read-protected, you must specify your A-disk read password.

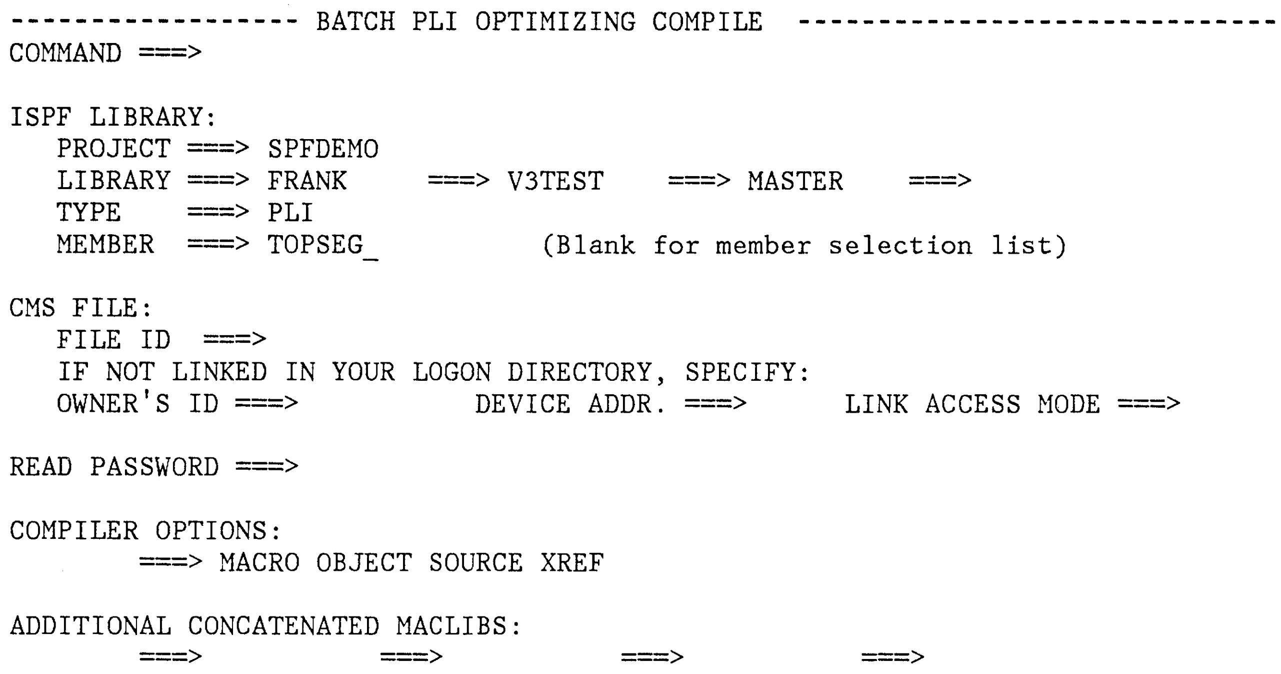
As with other secondary panels, you can bypass the batch selection panel by entering two numbers, separated by a decimal point, on the primary option menu. For example, enter "5.2" to select a batch COBOL compile. Note, however, that using this procedure does not allow you to verify or change the job statement parameters, nor does it allow you to generate multiple compilations (multiple job steps) within the same job.



```
----------------------- BATCH SELECTION MENU -------------------------------
OPTION ===> _

   1 - System assembler                     5 - PL/I optimizing compiler
   2 - OS/VS COBOL compiler                 6 - PASCAL/VS compiler
   3 - FORTRAN IV (G1) compiler             7 - Linkage editor
   4 - PL/I checkout compiler               8 - Load

BATCH JOB INFORMATION:  (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

YOUR A-DISK READ PASSWORD:
                    ===>
```

Figure 85. Batch Selection Panel

When job information has been entered and a language processor selected, an entry panel is displayed to allow you to specify ISPF library information or CMS filename, and processor options. Figure 86 shows an example for the PL/I Optimizing Compiler. Specify the project name, first library name, type qualifier, and member name to compile a program whose source is in an ISPF library. Specify the filename to compile a CMS file with a filetype of PLI or PLIOPT. (If you enter the CMS filename, it overrides any ISPF library specification.) The other parameters are optional.



```
------------------ BATCH PLI OPTIMIZING COMPILE  -----------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> FRANK     ===> V3TEST    ===> MASTER    ===>
   TYPE    ===> PLI
   MEMBER  ===> TOPSEG_         (Blank for member selection list)

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

READ PASSWORD ===>

COMPILER OPTIONS:
        ===> MACRO OBJECT SOURCE XREF

ADDITIONAL CONCATENATED MACLIBS:
        ===>           ===>           ===>           ===>
```

Figure 86. Batch - PL/I Optimizing Compiler Example

When you have filled in the panel and pressed the ENTER key, PDF generates the appropriate commands to be run in the batch machine. After PDF has generated the commands, it returns to the batch selection panel (unless you bypassed that panel by entering two numbers on the primary option menu), with a message "JOB STEP GENERATED" displayed in the short message area (line 1). If you bypassed the batch selection panel, PDF submits the generated commands and returns to the primary option menu. In either case, you may exit from the processor entry panel without generating any commands by entering the END command rather than pressing the ENTER key.

When the batch selection panel is redisplayed, the job statement parameters are shown for information only -- they are no longer intensified and may not be changed by overtyping. At this point, you 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 enter the END or RETURN command 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 corner of the screen. When you press the PA2 key, PDF returns to the the primary option menu.

The batch machine sends you messages indicating the progress of the job. When the job has completed, you can use the PDF reader utility (option 3.7) to retrieve TEXT and LISTING files produced by the batch job.

> **Note:** Erroneous results may occur if you use the batch option in split screen mode while you are also generating punch output from the other logical screen. Output from the two logical screens may be intermingled on the virtual punch if you jump back and forth between the screens.

ASSEMBLER/COMPILER PROCESSING

The entry panel shown in Figure 86 is typical of the batch language processor panels. In the figure, a library concatenation sequence of three libraries has been specified. The concatenation order is:

```
SPFDEMO.FRANK.PLI
SPFDEMO.V3TEST.PLI
SPFDEMO.MASTER.PLI
```

Before invoking the batch process, PDF scans 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) is created on your 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 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 PDF are erased at the completion of the batch process.

Notes:

1. The concatenation sequence serves two purposes: it is used by PDF 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. You can append other MACLIBs to the GLOBAL statement, following the ISPF libraries, by specifying them in the desired order in the "additional concatenated MACLIBs" fields.

3. The read password specified on the data entry panel must provide access to your A-disk as well as the source data. A minidisk with a read password of ALL is accessible regardless of the password specified on the panel.

4. The TEXT and LISTING files produced by the batch process are placed in your virtual reader using the DISK DUMP command. You may use the PDF reader utility (option 3.7) to retrieve these files following completion of the job.

5. If you specify the source as a CMS file (rather than an ISPF library), the filetype must be acceptable to the language processor (PLI or PLIOPT in this example).

6. 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 your own LOGON directory), you must provide the necessary LINK information: the user's id whose directory contains that information; the virtual device address of the minidisk on which the source resides; and the desired link access mode (e.g, RR).

7. The language processor options are passed to the processor exactly as you specify them. Only those options specified on the panel are passed to the language processor; PDF does not automatically generate any options.

The entry panels 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:

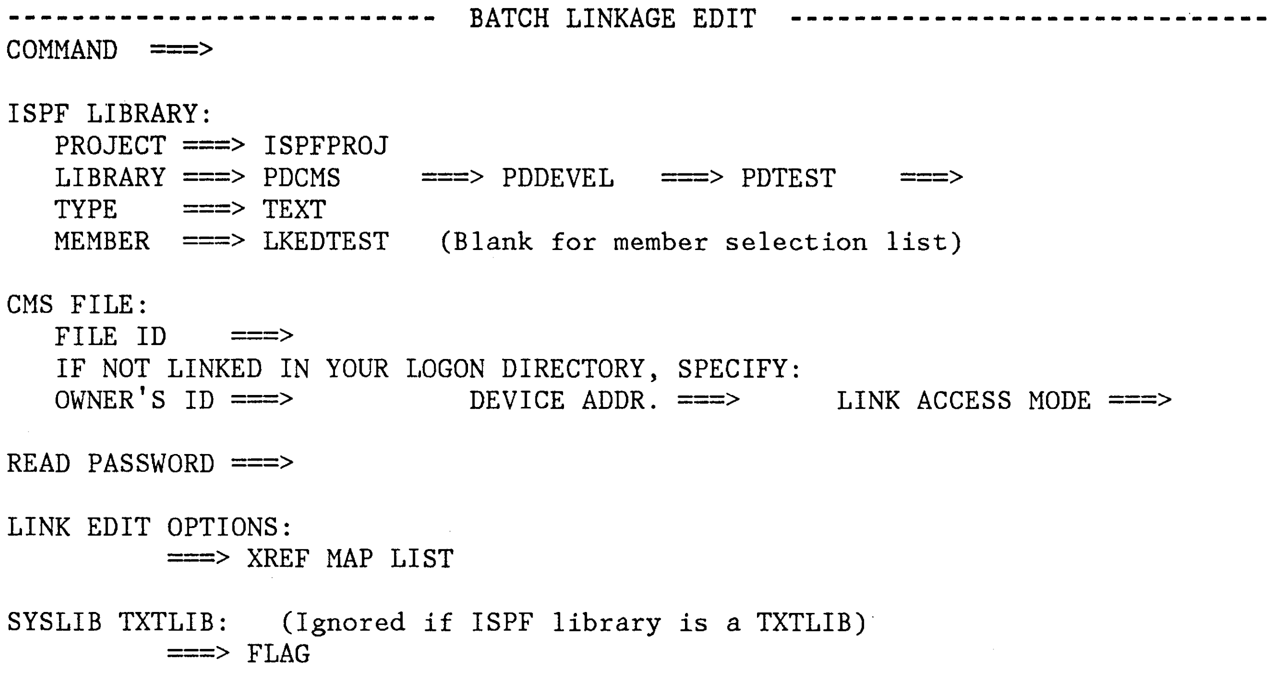
- When you specify the TEST option for the FORTRAN compiler, a special TEXT file is produced for later use by the FORTRAN interactive debug facility. The file is punched to your virtual reader, as is the generated FORTRAN file.

- When you specify the TEST option for the COBOL compiler, a SYSUT5 file is produced for later use by the COBOL interactive debug facility. The file is punched to your virtual reader.

- When you specify the OBJECT option 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 your virtual reader.

LINKAGE EDITOR PROCESSING

For the linkage editor, the CMS LKED command is invoked from the entry panel (Figure 87). ISPF generates the command from the input parameters you specify and then sends the command to the batch machine.



```
-------------------------- BATCH LINKAGE EDIT ------------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> ISPFPROJ
   LIBRARY ===> PDCMS     ===> PDDEVEL   ===> PDTEST    ===>
   TYPE    ===> TEXT
   MEMBER  ===> LKEDTEST   (Blank for member selection list)

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

READ PASSWORD ===>

LINK EDIT OPTIONS:
          ===> XREF MAP LIST

SYSLIB TXTLIB:   (Ignored if ISPF library is a TXTLIB)
          ===> FLAG
```

Figure 87. Foreground - Linkage Editor Example

The linkage editor input is contained in a single member of one of the specified ISPF libraries or in a CMS file with a filetype of "TEXT". Specifying a CMS file overrides the specification of a member of an ISPF library.

The input text libraries may be password protected; specify the password on the entry panel. It is not displayed, but is retained by PDF. All input libraries must have the same read password.

Additional text information may be included in a link edit if it is available in other text files or in a TXTLIB. Text information may reside in up to four ISPF libraries and in a TXTLIB that you link and access and then specify on the entry panel.

Note: If the you leave the "file-id" field blank and include an ISPF library that is defined as a TXTLIB, any TXTLIB that you specify in the "SYSLIB TXTLIB" field will be ignored. Since only one TXTLIB can be specified as SYSLIB input to the LKED command, you should ensure that any ISPF-defined input library is a set of sequential files, rather than a TXTLIB.

PDF does not automatically specify any linkage edit options. The options you specify on the panel are passed directly to the LKED command.

The LKED command produces two output files that may be disk dumped to you. It produces a file containing the load module that the linkage editor created, with the member name of the input file as the file name and a file type of LOADLIB. It also produces a listing file containing output according to the XREF, MAP, or LIST options you select. The file name of the listing file is the member name of the input file; the file type is LKEDIT. If you specify the linkage editor option NOPRINT, no listing file is produced. If you specify PRINT, the listing file is sent directly to the printer and will no longer exist. To retain the listing for browsing, specify the DISK option.

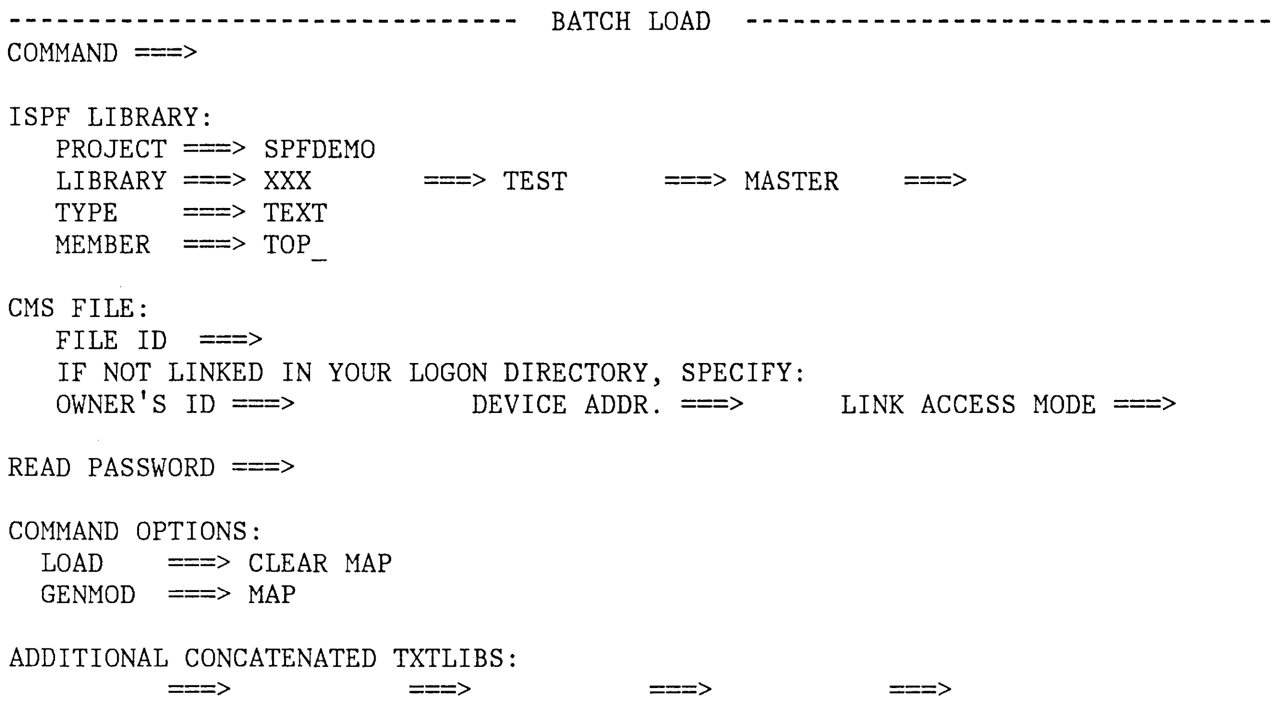
If you make entries in fields like those on the entry panel in Figure 87, some of the commands invoked might look like:

```
FILEDEF SYSLIB DISK PDCMS TXTLIB * (PERM
GLOBAL TXTLIB PDCMS PDDEVEL PDTEST FLAG
LKED LKEDTEST (XREF MAP LIST )
```

assuming that all the libraries are TXTLIBs. Note that only the first ISPF library is used as SYSLIB input.

LOAD PROCESSING

This batch option (option 5.6) provides an interface to the LOAD command (Figure 88). The entry panel allows a simple LOAD and GENMOD command to be issued with the specified options.

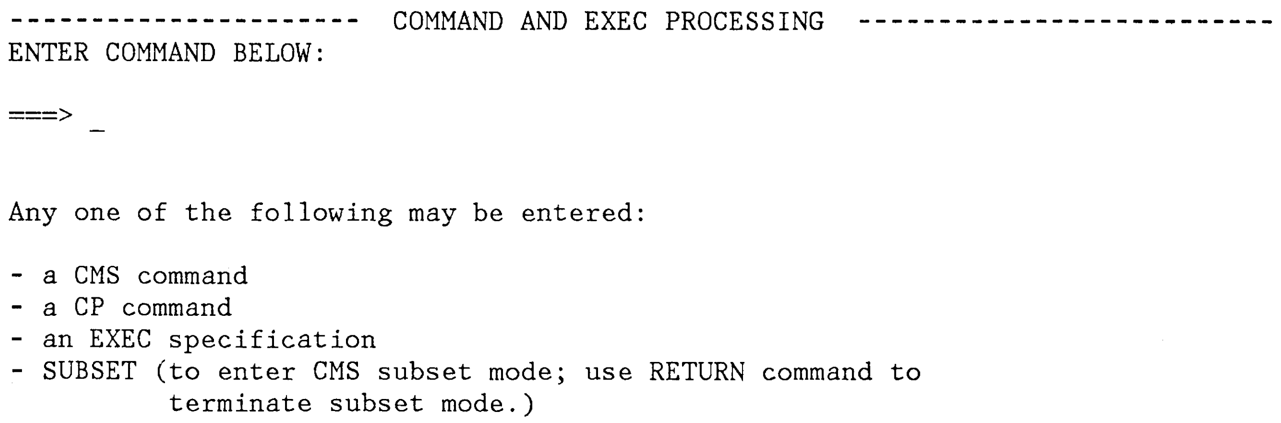


```
--------------------------------  BATCH LOAD  ---------------------------------
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===> SPFDEMO
   LIBRARY ===> XXX       ===> TEST      ===> MASTER    ===>
   TYPE    ===> TEXT
   MEMBER  ===> TOP_

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

READ PASSWORD ===>

COMMAND OPTIONS:
  LOAD    ===> CLEAR MAP
  GENMOD  ===> MAP

ADDITIONAL CONCATENATED TXTLIBS:
          ===>           ===>           ===>           ===>
```

Figure 88. Batch - LOAD Panel

CHAPTER 10. COMMAND (OPTION 6)

The command option allows direct entry of a CMS command, CP command, EXEC, or entry to CMS subset under PDF. When you select this option, the CMS command panel (Figure 89) is displayed.



```
--------------------- COMMAND AND EXEC PROCESSING -------------------------
ENTER COMMAND BELOW:

===> _


Any one of the following may be entered:

- a CMS command
- a CP command
- an EXEC specification
- SUBSET (to enter CMS subset mode; use RETURN command to
          terminate subset mode.)
```

Figure 89. CMS Command Panel

Note: PDF allows you to enter CMS and CP commands and EXECs in the command field of any panel. Option 6 also allows you to enter a long command that wraps around on the following line.

Enter the command or EXEC and its parameters to the right of the arrow, using standard CMS syntax. You may continue it on the next line, if necessary, to a maximum of 130 characters. You can also enter ISPF/PDF commands, such as END or RETURN, in this field. You need not enter "CMS" or "CP" before the command on this panel as you do on other panels. If you do, the executed command is not pre-entered when the command panel is redisplayed. If you enter "HELP", however, you will get the ISPF HELP information; to get the CMS HELP facility, enter:

```
===> CMS HELP xxxxx
```

You can enter CMS subset mode by entering "SUBSET" . In subset mode, you can enter more than one command at a time, and the PF and PA keys revert to their CMS definitions. Enter the RETURN command to exit from subset mode and return to PDF.

When you press the ENTER key, the command and any resulting output are displayed on the screen as in a normal CMS environment. (The return code presentation may appear slightly differently.) Following the last line displayed by the command, a "MORE..." message is displayed in the lower right corner of the screen. Press the PA2 or CLEAR key.

When you press the PA2 or CLEAR key, the panel is redisplayed. The command which was just executed is pre-entered to the right of the arrow. You may then enter another command, or enter the END command to return to the primary option menu.

Do not enter the following commands under PDF:

- LOGON, LOGOFF
- ISPSTART, PDF, and ISPF
- Commands that you are restricted from using by CMS or CP
- Commands requiring large parameter lists (130 characters is the maximum length, including command name)
- Commands that require the VM Communication Facility (VMCF)
- CMS/DOS environment commands
- VSAM commands
- ISPEXEC service calls

Results in all these cases are unpredictable.

> **Note:** When a CMS command or EXEC is executing under option 6, do not attempt to enter an HX (halt execution) command. This halts all commands or programs, including ISPF, bypassing the normal PDF termination procedure. To recover from this situation, a re-IPL of CMS is required.

CHAPTER 11. DIALOG TEST (OPTION 7)

The dialog test option provides users with aids for testing ISPF dialog parts (functions, panels, variables, messages, tables, skeletons) and complete ISPF applications. The dialog test option allows you to:

- Invoke selection panels, command procedures, and programs
- Display panels
- Add new variables and modify variable values
- Display a table's structure and status, and display, add, modify, and delete table rows
- Browse the ISPF log
- Execute dialog services
- Add, modify, and delete function and variable trace definitions
- Add, modify, and delete breakpoint definitions

You may use CMS DEBUG to complement this option if non-ISPF storage areas must be examined and manipulated.

When testing a dialog, you usually proceed in one of two ways:

1. Test individual dialog parts (panels, skeletons, messages, etc.) without invoking a function or a selection panel. Eventually, you terminate your test session by entering the END command on the dialog test primary option menu.

2. Test dialog functions (programs, commands, or selection panels) using the functions option. You can define traces and breakpoints before invoking the function.

When using the second approach, any requested traces for variable usage and dialog service calls are written to the ISPF log. The log may be browsed under dialog test.

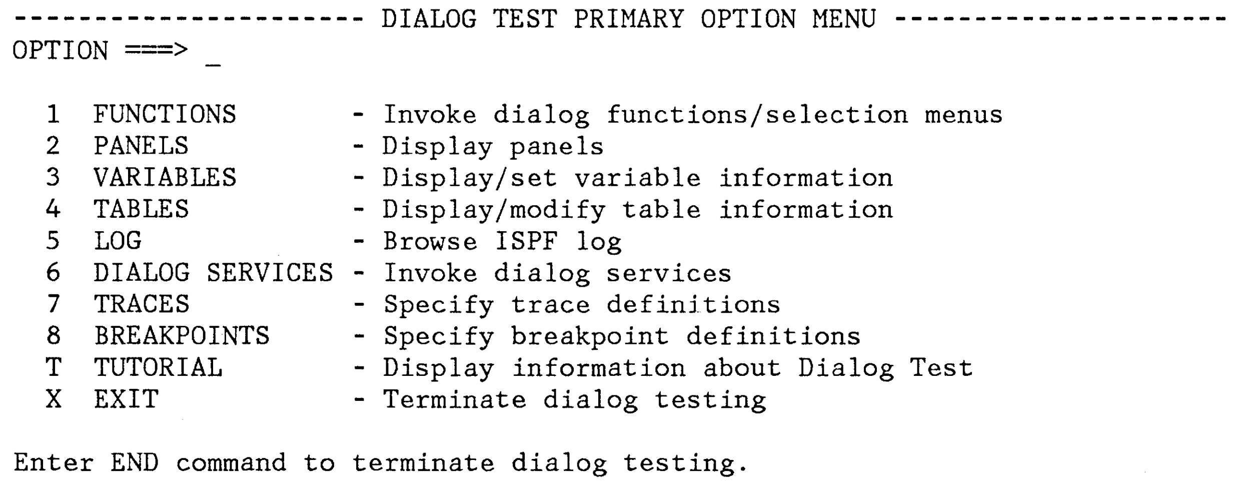
If you define a breakpoint, and the function gets to the breakpoint, dialog execution is suspended and dialog test displays the breakpoint primary option menu. At this point, you can access and manipulate dialog parts (variables, tables, etc.). Then, if you select the GO option from the breakpoint primary option menu, the dialog resumes execution.

At completion, you are returned to the invoke function option. If you select the CANCEL option from the breakpoint primary option menu, the dialog is cancelled and the first primary option menu that you were shown during your terminal session is redisplayed. For example, if the first screen displayed when you began your session was a master application menu (rather than the ISPF/PDF primary option menu), that master application menu would be redisplayed.

DIALOG TEST ENVIRONMENT

The dialog test panel (Figure 90) follows the conventions for a primary option menu. If you use the RETURN command from one of the selected dialog test options, you are always shown the dialog test primary option menu. If you use the END command from this panel, you return to the ISPF/PDF primary option menu.

When you enter dialog test, you enter a new user application with an application ID of ISR. All options listed on the dialog test primary option menu operate in this context. If you invoke a new function (using the functions option), a SELECT service call is performed, and the rules for the SELECT service are followed.



```
--------------------- DIALOG TEST PRIMARY OPTION MENU --------------------
OPTION ===> _

  1  FUNCTIONS        - Invoke dialog functions/selection menus
  2  PANELS           - Display panels
  3  VARIABLES        - Display/set variable information
  4  TABLES           - Display/modify table information
  5  LOG              - Browse ISPF log
  6  DIALOG SERVICES  - Invoke dialog services
  7  TRACES           - Specify trace definitions
  8  BREAKPOINTS      - Specify breakpoint definitions
  T  TUTORIAL         - Display information about Dialog Test
  X  EXIT             - Terminate dialog testing

Enter END command to terminate dialog testing.
```

Figure 90. Dialog Test - Primary Option Menu

Dialog test is itself a dialog and, therefore, uses dialog variables. Since it is important to allow your dialog to operate without interference (as though in a production environment), dialog test accesses and updates variables independently of your dialog variables.

All breakpoints and traces that you set in dialog test exist only for as long as you remain within the dialog test option.

You should always allocate the ISPF log; do not suppress allocation by specifying a log primary allocation of zero in ISPF/PDF option 0.2. Dialog test writes trace data to the log when you request it. In addition, if dialog test encounters an unexpected condition, problem data is written to the log.

When you enter dialog test, you are given the following ISPF facilities:

- All functions normally obtained by specifying the **TEST** parameter on the PDF command.

- The logging of all severe errors (both from user dialogs and dialog test). This is normally done when **TRACE** or **TRACEX** is specified on the PDF command.

- The cancellation of the logging of all ISPEXEC dialog service requests. Such logging normally occurs when TRACE or TRACEX is specified on the PDF command. You should use the dialog test trace options.

These facilities become active for both screens, and remain active as long as you remain in PDF, even after you leave dialog test.

The TEST, TESTX, TRACE, and TRACEX options are described in ISPF Dialog Management Services.

DIALOG TEST VARIABLE USAGE

When you select the dialog test option, you are given a new function pool, a new shared pool, and the ISRPROF profile. These pools are used if you set a variable, display a panel, call an ISPF service, etc. When you invoke a new dialog, dialog test uses the SELECT service, and follows the rules for the creation of new variable pools. For example, if you invoke a new dialog using the NEWPOOL option, dialog test creates new shared and function variables pools for you. The profile variable pool (ISRPROF) remains as it was.

Dialog variables should be initialized and set in the context of the dialog's processing. A dialog function's pools are created when it is invoked (when the SELECT is done). Therefore, to set variables in newly created pools, you must define a breakpoint early in your dialog's processing.

For example, if you invoke a dialog with the NEWPOOL parameter, you must define a breakpoint in that dialog to access that dialog's function and shared variable pools. You can modify the dialog's profile variable pool before invoking the dialog, since a new profile variable pool is not created.

When your dialog terminates, all variable pools created at dialog invocation are deleted.

DIALOG TEST SEVERE ERROR HANDLING

If your dialog encounters a severe error when it invokes a dialog service, that severe error is handled as requested by the dialog. The current CONTROL service ERRORS setting (CANCEL or RETURN; default is CANCEL) determines what is done. If CANCEL is in effect, when the error message panel is displayed, you may choose whether to continue dialog testing.

If you encounter a severe error when manipulating your dialog at a breakpoint, dialog test assumes that the CONTROL service ERRORS setting is CANCEL. For example, if you display a panel at a breakpoint and that panel is not found, the error message panel is displayed. This occurs even if your current dialog has an ERRORS setting of RETURN.

Regardless of the ERRORS setting, all of your severe errors are logged.

If dialog test encounters a severe error during its processing, the details are logged and the following message is shown to you on an error message display:

```
TEST SEVERE ERROR
DETAILS PRECEDE THIS MESSAGE IN THE ISPF LOG
```

Dialog test errors may occur because:

- Proper ISPF libraries are not being used.

- A programming problem has been encountered.

- You have attempted to execute dialog test recursively.

- You have invoked a dialog test option without being in test mode or without invoking dialog test first.

You should browse the ISPF log to determine the nature of the problem.

DIALOG TEST COMMANDS

PDF primary commands may be entered on dialog test panels. Five additional primary commands have special meaning during dialog test operations. These commands, and the dialog test options with which they function, are shown in the following table.

DIALOG TEST PRIMARY COMMAND	DIALOG TEST OPTION							
	3	4.1	4.3	4.4	4.5	7.1	7.2	8
CANCEL	X		X	X		X	X	X
END	X	X	X	X		X	X	X
LOCATE	X	X	X	X	X	X	X	X
QUAL								X
RESUME								X

Dialog test has three line commands that have special meaning during testing operations. These commands, and the options with which they function, are shown in the following table.

DIALOG TEST LINE COMMAND	DIALOG TEST OPTION					
	3	4.3	4.4	7.1	7.2	8
D (Delete)		X	X	X	X	X
I (Insert)	X	X	X	X	X	X
R (Repeat)		X	X	X	X	X

In using the dialog test primary and line commands, you should be aware of the following:

- Both a primary command and line commands may be specified before you press the ENTER key.

- You can enter multiple line commands on the display.
- You cannot perform a deletion if one of the included lines contains another line command.
- You can delete lines with any kind of input error.
- The line commands are processed in row order when you press the ENTER key. Any fields changed in that row are handled before a line command is processed.
- A primary command is handled after processing for all line commands has been completed.
- A number may be specified with each line command, as in the editor, to denote repetitive operation.

Primary Commands

The following primary commands have special meaning during dialog test operations. They are entered in the command line of the applicable dialog test option panels.

CANCEL (CAN) Ends the current option. Any changes made to the data are ignored.

END Ends the current option. Any changes made to the data now take effect.

LOCATE (LOC, L) argument Positions the scrollable display to the first (next) row containing the argument. The scan starts at the end of the first row currently being displayed. A message is displayed indicating the result of the scan.

QUAL This command is entered only from the breakpoints panel. It displays the breakpoint qualification data to the user.

RESUME (RES) This command is entered on the qualification panel. It restores the format of the breakpoints panel. Each breakpoint that has qualification is flagged by the characters "*QUAL*" in columns 75 to 80 on that line of the breakpoints panel.

Line Commands

The following line commands have special meaning during testing operations:

D or Dn Delete one line or n lines starting with this line.

I or In Insert one line or n lines directly after this line (with underscores and quotes in the appropriate fields).

R or Rn Repeat this line once or n times.

FUNCTIONS (OPTION 7.1)

The functions option allows you to test a dialog function (panel, command procedure, or program) without having to build supporting code or panels. The name of the dialog function and the parameters that may be passed are the same as those that can be specified (from a dialog function) when invoking the SELECT service. When you press the ENTER key, a SELECT is done.

If, at a breakpoint, you invoke a new function or selection panel, the previous one is suspended and the new one executed. When the new function or selection panel completes execution, you are returned to the invoke function entry panel. The old function resumes execution when you enter the END command.

When the originally invoked function completes execution, the invoke dialog function panel is redisplayed.

When you select this option, a panel is displayed that allows you to identify the dialog function that you want to test (Figure 91).

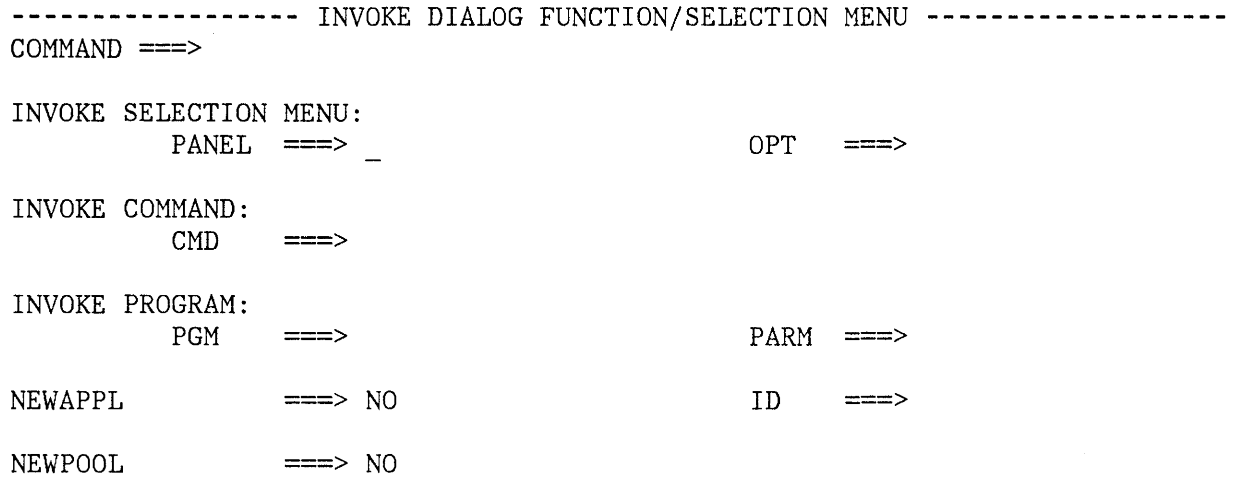
- To invoke a selection panel, use the following fields, under the heading **INVOKE SELECTION MENU:**

  PANEL The name of the selection panel to be displayed.

  OPT An optional parameter indicating an initial selection option that must be valid from the specified selection panel.

- To invoke a command, use the following field, under the heading **INVOKE COMMAND:**

  CMD The name of a command procedure (EXEC), or any CMS command, to be invoked as a dialog function. You may include command parameters.



```
------------------ INVOKE DIALOG FUNCTION/SELECTION MENU ------------------
COMMAND ===>

INVOKE SELECTION MENU:
          PANEL  ===> _                       OPT   ===>

INVOKE COMMAND:
          CMD    ===>

INVOKE PROGRAM:
          PGM    ===>                         PARM  ===>

NEWAPPL          ===> NO                      ID    ===>

NEWPOOL          ===> NO
```

Figure 91. Dialog Test - Invoke Function Entry Panel

- To invoke a program, use the following fields, under the heading **INVOKE PROGRAM:**

  PGM The name of a program to be invoked as a dialog function.

  PARM Optional parameters to be passed to the program.

You can also specify:

NEWAPPL Indication of whether a new application is being invoked.

yes, y This function is a new application.

no, n This function is not a new application.

ID A 1- to 4-character application ID for a new application. If you invoke a new application and leave the ID field blank, the default ID of ISP is used. Note that the ID determines the names of the profile and the command table to be used for the application.

NEWPOOL Indication of whether a new shared variable pool is to be created. This value is ignored if the value for NEWAPPL is "yes" or "y."

- **yes, y** Create a new shared variable pool.
- **no, n** Do not create a new shared variable pool.

Be aware of the following items when using the functions option:

OPT and PARM The OPT and PARM parameter input fields each continue on the following line.

REQUIRED FIELDS To invoke a function, you must specify a value for either the PANEL, CMD, or PGM parameter; but you may not specify more than one of them.

PANELS (OPTION 7.2)

The panels option allows you, during panel development, to test newly created or modified panels and messages without having to build supporting code to display them. Any variables referenced and set during panel processing are handled according to standard ISPF protocol. When you select the panels option, you are shown a panel (Figure 92).

On the panel are the following fields:

PANEL NAME The name of the panel to be displayed.

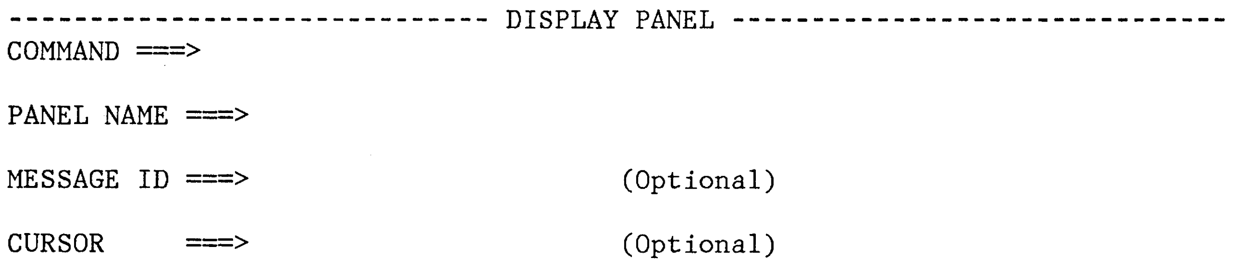
MESSAGE ID The identifier of a message to be displayed on the panel (optional).

CURSOR The name of the field on the panel where the cursor is to be positioned (optional).

These are the same parameters that may be specified (from a dialog function) when invoking the DISPLAY service.

If you specify a message id without identifying a panel name, the message is displayed on a separate panel (Figure 93). The long message portion of the identified message is displayed when you enter the HELP command on that panel.

When you use the END command from the panel being tested, the display panel entry panel is redisplayed.



```
----------------------------- DISPLAY PANEL -----------------------------
COMMAND ===>

PANEL NAME ===>

MESSAGE ID ===>                         (Optional)

CURSOR     ===>                         (Optional)
```

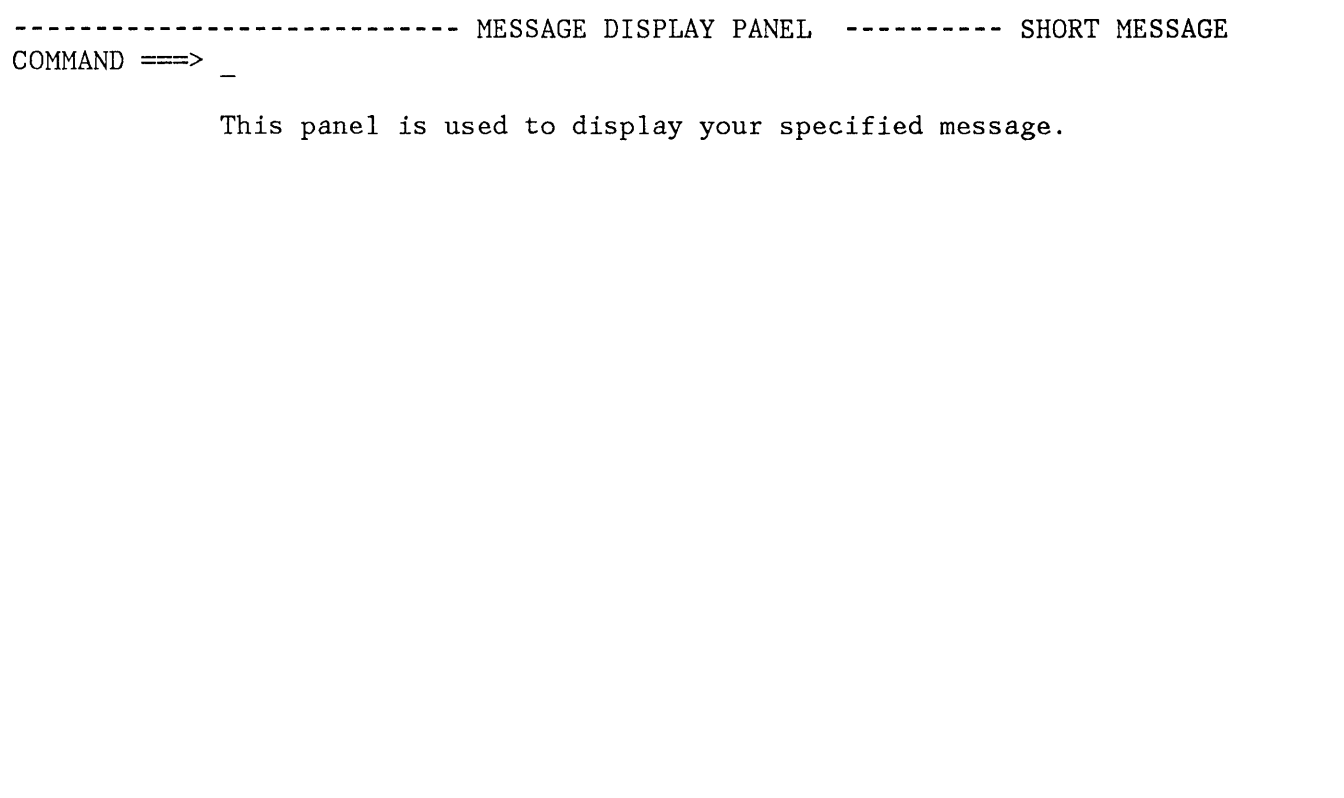
Figure 92. Dialog Test - Display Panel Entry Panel

Be aware of the following points when using the panels option:

VARIABLES You may set variables using the variables option (option 3) before you display the panel. When you display the panel, you can enter new or modified data and then verify the variables using option 3 again.

PANEL EXECUTION When the panel is displayed, the)INIT and)PROC sections of the displayed panel are processed in the same way they would be by the DISPLAY service.

DATA RETENTION Data that you enter on the panel you display is retained until you change it or until you leave the display panel option.



```
---------------------------- MESSAGE DISPLAY PANEL  ---------- SHORT MESSAGE
COMMAND ===> _

             This panel is used to display your specified message.
```

Figure 93. Dialog Test - Message Display Panel

VARIABLES (OPTION 7.3)

The variables option allows you to:

- Display all the ISPF variables defined in the dialog application you are testing.

- Change the value of a variable by overtyping it.

- Define new variables by inserting lines or by changing the name or pool of a listed variable.

When you select this option, a scrollable display (Figure 94) indicates all the current variables for the dialog being tested.

Each line of the display represents a variable and contains:

''''   Line command area.

VARIABLE Name of the variable: an alphameric 8-character name, with the first character alphabetic. The alphabetic characters are translated to uppercase when you press the ENTER key. This field is required.

P Pool in which the variable exists, a required one-character field:

- **F** Function variable pool
- **S** Shared variable pool
- **P** Profile variable pool

A Attributes of the variable, a nonmodifiable one-character field:

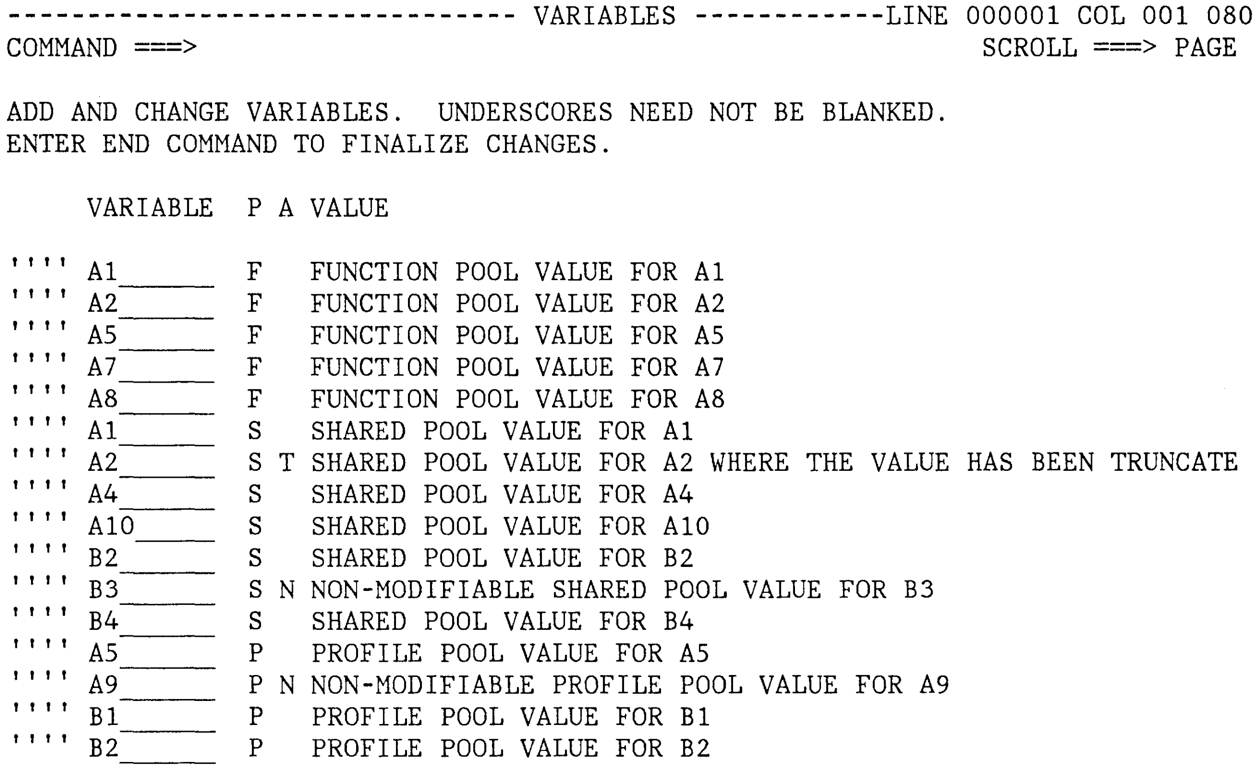
- **N** Nonmodifiable variable. Some system-reserved variables are not modifiable.
- **T** Truncated variable value. The actual value is longer than the 58 characters displayable on the panel.

  Note: If you change a line showing a truncated value, only the data shown on the display is stored as the new value.

VALUE Value of the variable up to 58 characters.

The variables option uses the CANCEL, END, and LOCATE dialog test primary commands, and the I dialog test line command. These commands are described in the section entitled "Dialog Test Commands."

Normally, the variable pools are updated with the data from the display when you use the END command to leave the option.



```
------------------------------- VARIABLES ------------LINE 000001 COL 001 080
COMMAND ===>                                                SCROLL ===> PAGE

ADD AND CHANGE VARIABLES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

     VARIABLE  P A VALUE

'''' A1______  F   FUNCTION POOL VALUE FOR A1
'''' A2______  F   FUNCTION POOL VALUE FOR A2
'''' A5______  F   FUNCTION POOL VALUE FOR A5
'''' A7______  F   FUNCTION POOL VALUE FOR A7
'''' A8______  F   FUNCTION POOL VALUE FOR A8
'''' A1______  S   SHARED POOL VALUE FOR A1
'''' A2______  S T SHARED POOL VALUE FOR A2 WHERE THE VALUE HAS BEEN TRUNCATE
'''' A4______  S   SHARED POOL VALUE FOR A4
'''' A10_____  S   SHARED POOL VALUE FOR A10
'''' B2______  S   SHARED POOL VALUE FOR B2
'''' B3______  S N NON-MODIFIABLE SHARED POOL VALUE FOR B3
'''' B4______  S   SHARED POOL VALUE FOR B4
'''' A5______  P   PROFILE POOL VALUE FOR A5
'''' A9______  P N NON-MODIFIABLE PROFILE POOL VALUE FOR A9
'''' B1______  P   PROFILE POOL VALUE FOR B1
'''' B2______  P   PROFILE POOL VALUE FOR B2
```

Figure 94. Dialog Test - Variables Entry Panel

Manipulating Variables

The rows of the display are ordered by the pool containing the variables, then alphabetically by variable name within each pool. The function variable pool is listed first, followed by the shared variable pool, and then the profile variable pool. Insertions are left where they are entered on the display.

Modifications to the display are processed when you press the ENTER key. Updating of the variable pools occurs when you enter the END command.

You can create new dialog variables, but you cannot create two variables with the same name in the same variable pool. Also, you cannot delete a variable, but you can set its value to nulls.

To create a new variable, you may either:

- Use the I (insert) line command to create a new row and then enter the variable name, pool, and value on the new line. For each field, move the cursor to the start of the field and enter new information. The underscores are pad characters and need not be blanked out.

- Take an existing variable and change its name, its pool, or both. This creates a new variable with the old variable's last value and resets the old variable's value to nulls.

By using the second method, you can interchange the values of two or more variables by simply changing their names. For example, you can interchange the values for variables A and B by changing the variable name A to B and name B to A and then pressing ENTER.

Variables Usage Notes

When using the variables option, you should be aware of the following:

TEST MODE Variable manipulations performed under dialog test at a breakpoint are considered an extension of your dialog and so are handled in user mode. Dialog variables, table data, and service return codes that you introduce, delete, or change are treated as though your dialog had made those changes.

VARIABLE LIFE Profile variables that you create remain in your profile pool from one dialog test session to another. Shared and function variables exist only for the duration of dialog test.

VARIABLE SEQUENCE Any changes you make on the scrollable display are left where you enter them. Sorting is performed only when you reselect the variables option.

SPLIT-SCREEN MODE In split screen mode, a profile variable pool may be shared between two logical screens. Since the variables option takes a snapshot of the variables, any change to a profile variable on one screen is not immediately reflected on the other screen. To get the latest changes, reselect the variables option. Also, when one profile variable is changed on two logical screens using split screen mode, the changed profile variable on the screen where the last END command was entered takes precedence.

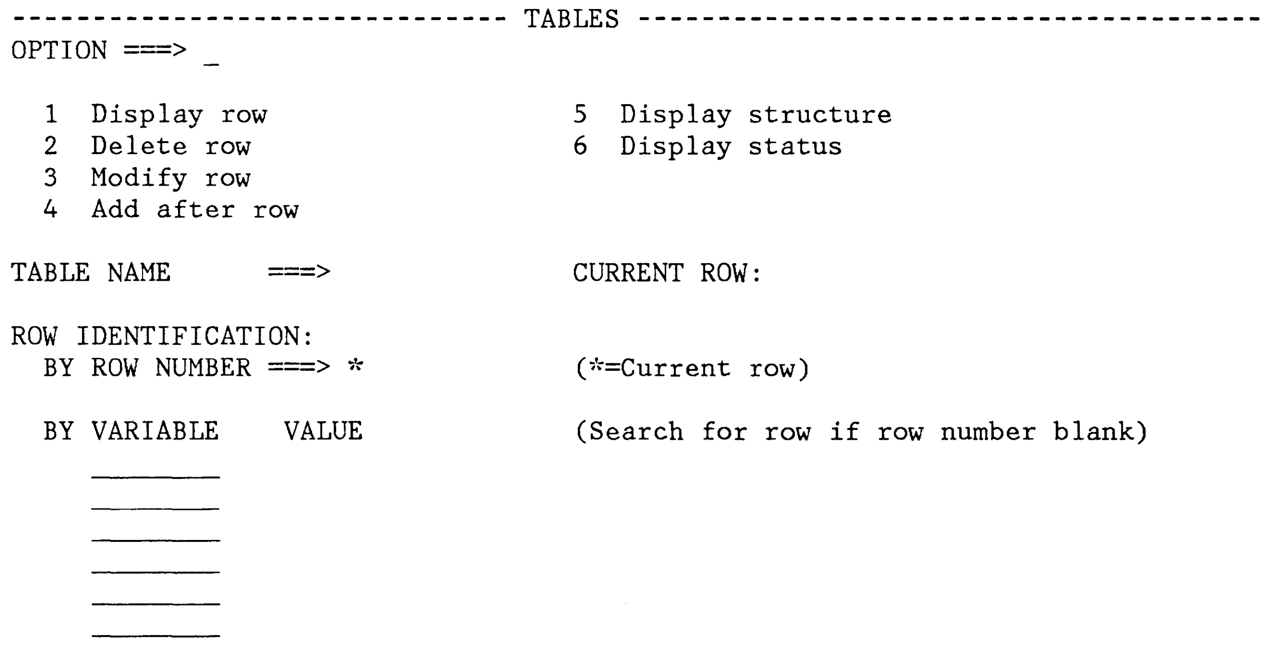
VARIABLE VALUE Variables defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command. When a VDEFINE error occurs, a panel identifies the data and its value and describes the error. You must then correct the error and press the ENTER key. If you create a new variable by changing the pool indicator of an existing variable defined as other than character, the new variable will have character format.

HEXADECIMAL DATA

Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.

TABLES (OPTION 7.4)

The tables option allows you to examine and manipulate the rows of a table, and to display table structure and status. When you select this option, a panel is displayed (Figure 95) on which you indicate the table function you want and the parameters needed to identify the table.



```
------------------------------ TABLES --------------------------------------
OPTION ===> _

  1  Display row                    5  Display structure
  2  Delete row                     6  Display status
  3  Modify row
  4  Add after row

TABLE NAME      ===>                CURRENT ROW:

ROW IDENTIFICATION:
  BY ROW NUMBER ===> *              (*=Current row)

  BY VARIABLE    VALUE              (Search for row if row number blank)
     ________
     ________
     ________
     ________
     ________
     ________
```

Figure 95. Dialog Test - Tables Selection Panel

Use the fields on the tables entry panel as follows:

OPTION Enter the number of the table function:

1 Display row

2 Delete row

3 Modify row

4 Add after row

5 Display structure

6 Display status

TABLE NAME Enter the name of the table you are interested in. The table must be open for all but the display status option (tables option 6). You can use dialog test option 6 (invoke dialog service) to open the table if necessary.

CURRENT ROW Displays the position number of the current row after you have identified a table. This field is nonmodifiable.

ROW IDENTIFICATION Identify a row, either directly by row number or indirectly by specifying a search argument of table variable names and their values.

BY ROW NUMBER Enter the position number of the table row desired or * for the current row. If you are adding a row, you can use:

top, t, 0 Make the new row first in the table.

bottom, bot, b Make the new row last in the table.

BY VARIABLE Overtype the underscores beneath this heading with the names of variables in the table whose values are to be used to search the table for a row with matching contents.

VALUE Enter the value to use in the search, up to 58 characters. For an abbreviated search, enter the beginning characters followed by an asterisk (*).

Once you have specified a table name, it is retained until you change it or until you leave dialog test.

You must specify the row number or a search argument list (using the BY VARIABLE and VALUE fields) for options 1, 2, 3, or 4 to identify the row of interest. Specifying neither while leaving the * as the row number implies the current row. If you specify both a row number and a search argument, the row number is used and the search argument is ignored. The row number may be blank; numeric; an asterisk (*) for the current row; top, t, or 0 (numeric zero) for adding a row as first in the table; or bottom, bo, or b for adding a new row as last in the table. When a row is added, the new row is placed after the row identified.

The current row pointer in the table can be changed only at your request or by your dialog.

The search argument list consists of variable names and values that let you specify the values that specific variables have in the row of interest. You can specify the entire value, abbreviate the value with an asterisk (to find the row containing a variable beginning with specified characters), or leave the row blank. The search begins with the row following the current row. If a row matching the search argument is not found, the current row pointer is set to the top and the search may be repeated.

Display Row (Tables Option 1)

The display row option is used to display the contents of an existing row in an open table. It is selected when you enter 1 on the option line of the tables panel, after naming an open table and identifying an existing row in the table.

You are shown a scrollable display of the table row data (Figure 96). In the figure, the variables constitute one table row. Each line on the display indicates:

VARIABLE Variable name

T Type of variable:

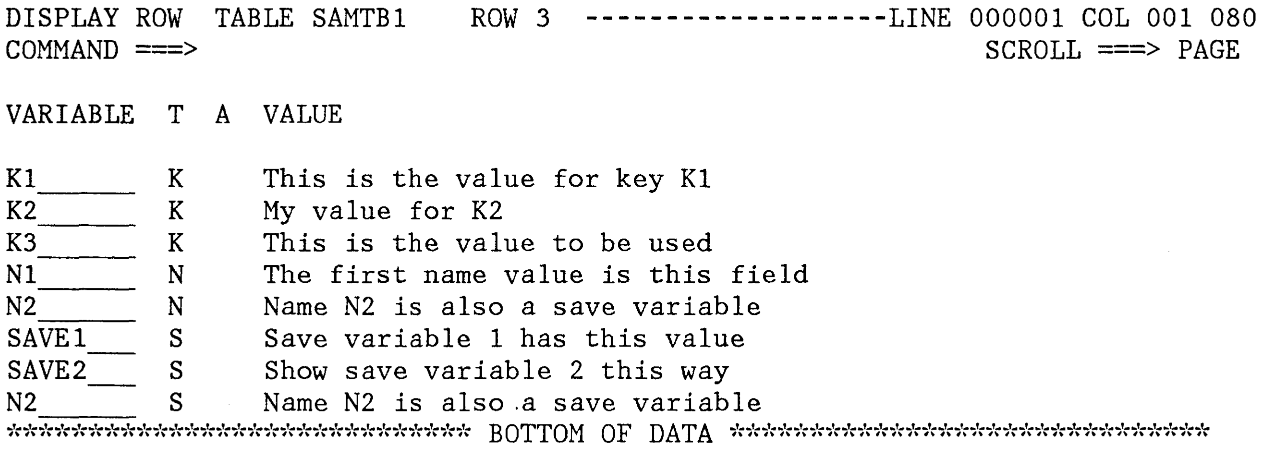
- **K** Key variable
- **N** Name (non-key) variable
- **S** Save (extension) variable

A Attribute of each variable:

- **T** Truncated to 58 characters for display

VALUE Value of the variable up to 58 characters

The display row option uses the END and LOCATE dialog test primary commands. These commands are described in the section entitled "Dialog Test Commands."



```
DISPLAY ROW  TABLE SAMTB1    ROW 3  -------------------LINE 000001 COL 001 080
COMMAND ===>                                               SCROLL ===> PAGE

VARIABLE  T  A  VALUE

K1______  K     This is the value for key K1
K2______  K     My value for K2
K3______  K     This is the value to be used
N1______  N     The first name value is this field
N2______  N     Name N2 is also a save variable
SAVE1___  S     Save variable 1 has this value
SAVE2___  S     Show save variable 2 this way
N2______  S     Name N2 is also a save variable
****************************** BOTTOM OF DATA *******************************
```

Figure 96. Dialog Test - Display Table Row Entry Panel

Delete Table Row (Tables Option 2)

The delete table row option is used to remove an existing row from an opened table. It is selected when you enter 2 on the option line of the tables panel after naming an open table and identifying an existing row of the table.

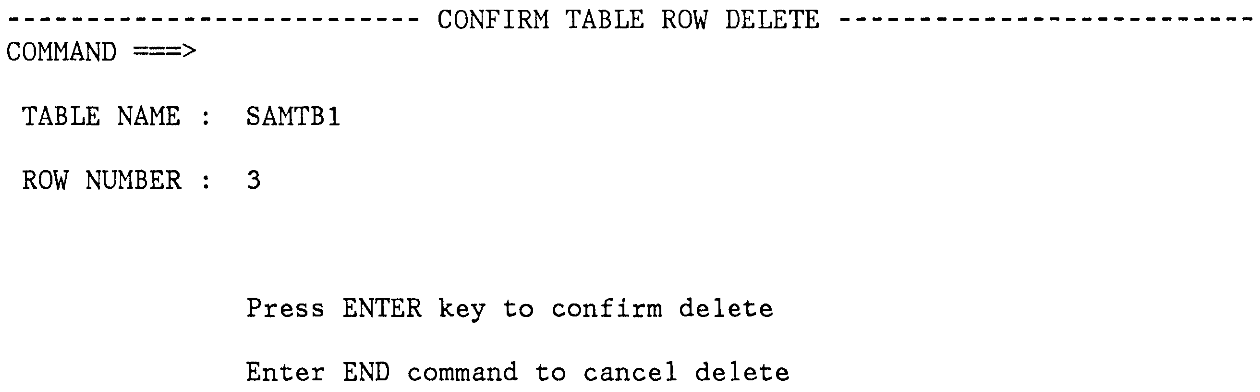
When you select this option, you are shown a panel (Figure 97) asking you to confirm the delete request (as in the PDF delete function).

The fields on the panel are:

TABLE NAME Name of an open table

ROW NUMBER Number of the row to be deleted

Press the ENTER key to delete the row, or enter the END or CANCEL command to cancel the deletion.



```
------------------------- CONFIRM TABLE ROW DELETE -------------------------
COMMAND ===>

 TABLE NAME :  SAMTB1

 ROW NUMBER :  3


               Press ENTER key to confirm delete

               Enter END command to cancel delete
```

Figure 97. Dialog Test - Delete Table Row Confirmation Panel

Modify Row (Tables Option 3)

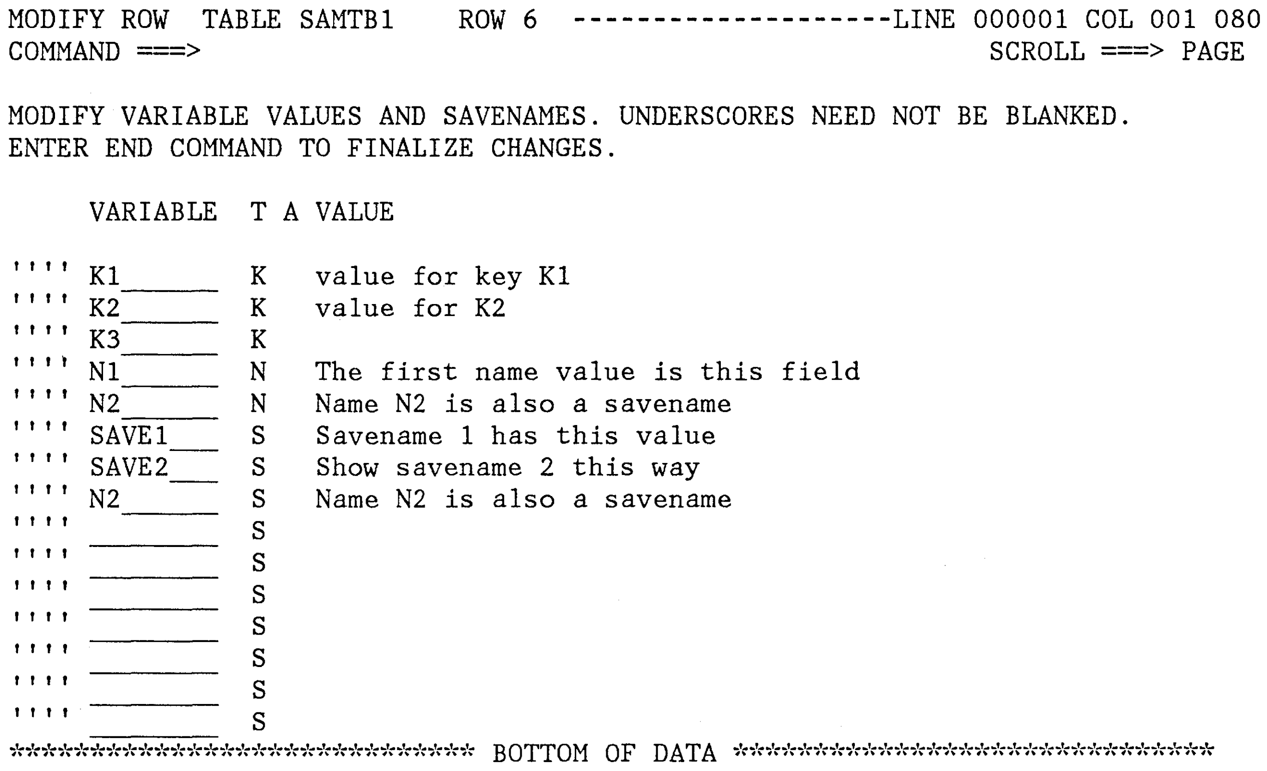
The modify row option is used to change the contents of an existing row of an opened table. It is selected when you enter 3 in the option field of the tables panel after naming an opened table and identifying an existing row of the table.

When you select this option, a scrollable display (Figure 98) is shown. In this example, row 6 of table SAMTB1 is displayed for modification. Row 6 contains eight variables.

Each line on the panel represents a variable in the row of the table, and contains:

'''' Line command area

VARIABLE Variable name, modifiable only for save variables



```
MODIFY ROW  TABLE SAMTB1    ROW 6  --------------------LINE 000001 COL 001 080
COMMAND ===>                                                SCROLL ===> PAGE

MODIFY VARIABLE VALUES AND SAVENAMES. UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

     VARIABLE  T A VALUE

'''' K1______  K   value for key K1
'''' K2______  K   value for K2
'''' K3______  K
'''' N1______  N   The first name value is this field
'''' N2______  N   Name N2 is also a savename
'''' SAVE1___  S   Savename 1 has this value
'''' SAVE2___  S   Show savename 2 this way
'''' N2______  S   Name N2 is also a savename
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
****************************** BOTTOM OF DATA ******************************
```

Figure 98. Dialog Test - Modify Table Row Entry Panel

T Type of variable, nonmodifiable:

- **K** Key variable
- **N** Name (non-key) variable
- **S** Save (extension) variable

A Attribute of each variable, nonmodifiable:

- **T** Truncated to 58 characters for display

VALUE Value of the variable up to 58 characters

Enter or change the values for the key, name, and save variables in the VALUE column. Enter new save variables by overtyping the underscores in the NAME column with the variable names and specifying the desired values. (The underscores are pad characters and need not be blanked out.)

When using the modify table row option, be aware of the following:

- You cannot modify a row in a keyed table unless the keys are unique among all the existing rows of the table. If you try to modify a row to have a non-unique key, you are given an opportunity to correct your error.

- You may not change the variable name for a key variable or name variable; if you do, an error message is displayed and the original name is restored.

- You may not delete a key or name variable and its value from the display or table row.

- If you delete a save variable, it is assumed that the variable no longer exists in this row.

- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable (closest to the bottom of the scrollable display).

- If the table has keys, the values for the keys in the added row must be unique among the existing rows when you leave the modify row option. Otherwise, a message is displayed and the row is redisplayed so that you can change the keys.

- Blank save names are ignored and need not be deleted, even if data is left in the value.

- Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.

- Variables that have been defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command.

- When you leave the modified row option using the END command, the row is replaced and the message "ROW MODIFIED" is issued.

The modify row option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. These commands are described in the section entitled "Dialog Test Commands." Inserted and repeated lines always have a type of S since only save variables may be added to a row of an existing table. Only save variables may be deleted.

Add After Row (Tables Option 4)

The add after row option is used to add a new row after a selected row of an opened table. It is selected when you enter 4 on the option line of the tables panel after naming an open table and identifying the row of the table that the new row will follow.

When you select this option, a scrollable display is shown (Figure 99), containing all the key and name variables in the table.

Each row of the display contains:

'''' Line command area

VARIABLE Variable name

T Type of variable, nonmodifiable:

K Key variable

N Name (non-key) variable

S Save (extension) variable

A Attribute of each variable, nonmodifiable (not used for this option)

VALUE Space for the variable value to be added, up to 58 characters

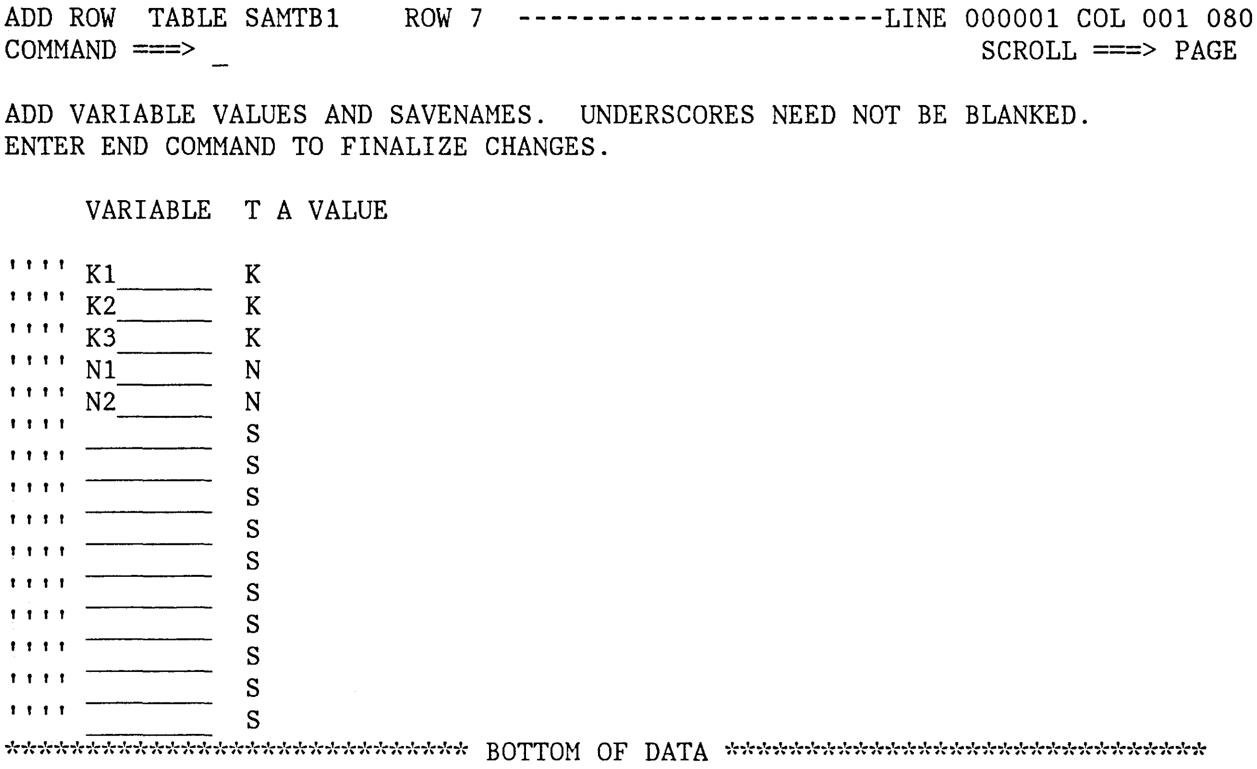
Enter the values for the key and name variables in the VALUE column, which is originally initialized to all nulls. You cannot change the names of the key and name variables because they were established when the table was created.

You can enter save variables, identified by TYPE S, by overtyping the underscores with the save variable names and specifying the desired values. (The underscores are pad characters and need not be blanked out.)

You can add a row with no values to the table but you will be asked to confirm such an action to guard against inadvertent use of the END command.

When using the add after row option, be aware of the following points:

- You cannot add a row in a keyed table unless the keys are unique among all the existing rows of the table. If you try to modify a row to have a non-unique key, you will be given an opportunity to correct your error.



```
ADD ROW  TABLE SAMTB1    ROW 7  ----------------------LINE 000001 COL 001 080
COMMAND ===> _                                              SCROLL ===> PAGE

ADD VARIABLE VALUES AND SAVENAMES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

     VARIABLE  T A VALUE

'''' K1______  K
'''' K2______  K
'''' K3______  K
'''' N1______  N
'''' N2______  N
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
'''' ________  S
****************************** BOTTOM OF DATA ******************************
```

Figure 99. Dialog Test - Add After Row Entry Panel

- You may not delete a key or name variable and its value from the display or table row.

- You may not change the variable name for a key or name; if you do, an error message is displayed and the original name is restored.

- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable (closest to the bottom of the scrollable display).

- If the table has keys, the values for the keys in the added row must be unique among the existing rows when you leave the add after row option. Otherwise, a message is displayed and the row is redisplayed so that you can change the keys.

- Blank save names are ignored and need not be deleted, even if data is left in the value.

- Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.

- Variables that have been defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes an error message to be displayed when you use the END command.

The add after row option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. Refer to "Dialog Test Commands." Inserted and repeated lines always have a type of S since only save variables may be added to a row of an existing table. Only save variables may be deleted.

Display Structure (Tables Option 5)

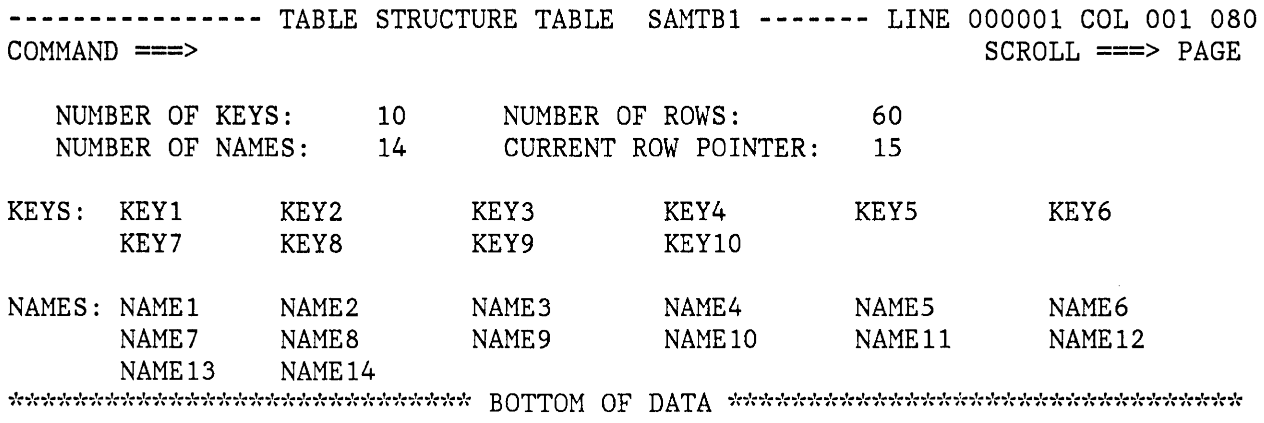
When you select option 5 from the tables option panel, you are shown a scrollable display (Figure 100) with the table name (in this case, SAMTB1) in the panel header.

The display shows the following fields:

NUMBER OF KEYS	Number of key variables in a row
NUMBER OF NAMES	Number of name variables in a row
NUMBER OF ROWS	Number of rows currently in the table
CURRENT ROW POINTER	Current row pointer value
KEYS	A list of the names of all the key variables
NAMES	A list of the names of all the name variables

The KEYS and NAMES lists are scrollable, and you can use the LOCATE command to find a specific variable name.

The display structure option uses the LOCATE dialog test primary command. Refer to "Dialog Test Commands."



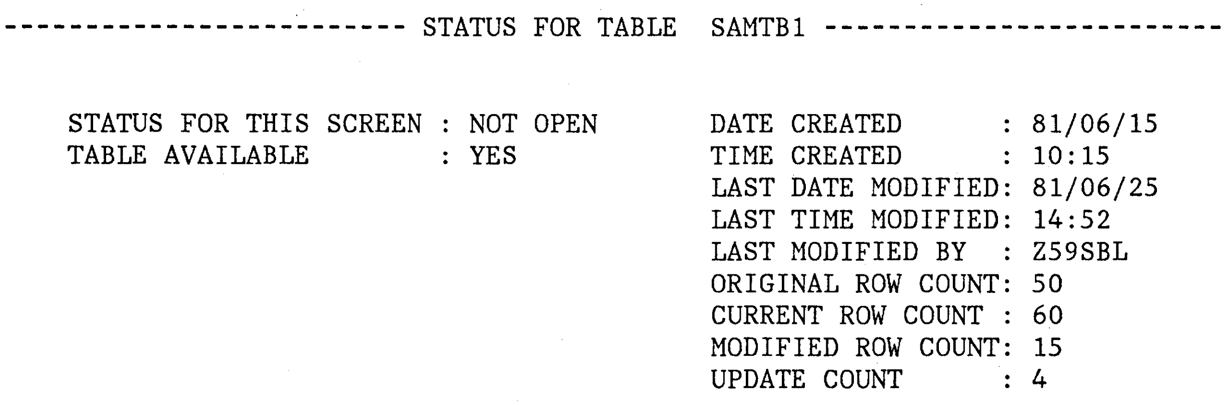
```
---------------- TABLE STRUCTURE TABLE  SAMTB1 ------- LINE 000001 COL 001 080
COMMAND ===>                                                 SCROLL ===> PAGE

   NUMBER OF KEYS:      10      NUMBER OF ROWS:        60
   NUMBER OF NAMES:     14      CURRENT ROW POINTER:   15

KEYS:  KEY1      KEY2        KEY3        KEY4        KEY5        KEY6
       KEY7      KEY8        KEY9        KEY10

NAMES: NAME1     NAME2       NAME3       NAME4       NAME5       NAME6
       NAME7     NAME8       NAME9       NAME10      NAME11      NAME12
       NAME13    NAME14
***************************** BOTTOM OF DATA ********************************
```

Figure 100. Dialog Test - Table Structure Display

Display Status (Tables Option 6)

If you select option 6 from the tables option panel, one of two data information panels is displayed. The information reflects all operations using the specified table, including those done at your request by the table options under dialog test.

If the table is not open for your user id, you are shown an information panel (Figure 101).



```
------------------------ STATUS FOR TABLE  SAMTB1 ------------------------

    STATUS FOR THIS SCREEN : NOT OPEN       DATE CREATED      : 81/06/15
    TABLE AVAILABLE        : YES            TIME CREATED      : 10:15
                                            LAST DATE MODIFIED: 81/06/25
                                            LAST TIME MODIFIED: 14:52
                                            LAST MODIFIED BY  : Z59SBL
                                            ORIGINAL ROW COUNT: 50
                                            CURRENT ROW COUNT : 60
                                            MODIFIED ROW COUNT: 15
                                            UPDATE COUNT      : 4
```

Figure 101. Dialog Test - Table Status Display for a Closed Table

The panel indicates the following information:

STATUS FOR THIS SCREEN Whether the table is OPEN or NOT OPEN for this logical screen

TABLE AVAILABLE YES or NO, whether you can open the table

DATE CREATED Date the table was created

TIME CREATED Time the table was created

LAST DATE MODIFIED Date the table was last modified

LAST TIME MODIFIED Time the table was last modified

LAST MODIFIED BY User id of the user who last modified the table

ORIGINAL ROW COUNT The number of rows that were added to a newly created table before closing the table for the first time.

CURRENT ROW COUNT The number of rows currently in the table.

MODIFIED ROW COUNT The number of rows in the table that have been modified at least once. A row that has been added to an existing table is also considered a modified row.

UPDATE COUNT Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

> **Note:** Tables option 4.3 (modify row) allows you to modify a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when modifying a key value.

If the table is open for your user id, you are shown an information panel (Figure 102).

This panel indicates the following information:

STATUS FOR THIS SCREEN Whether the table is OPEN or NOT OPEN for this logical screen

OPEN OPTION Option used to open the table (WRITE, NOWRITE, SHR WRITE, or SHR NOWRITE)

TABLE ON DISK Whether the table has been saved on disk, YES or NO

LAST TABLE SERVICE Name of the last table service invoked

LAST SERVICE RETURN CODE Last table services return code

CURRENT ROW POINTER Current position in the table

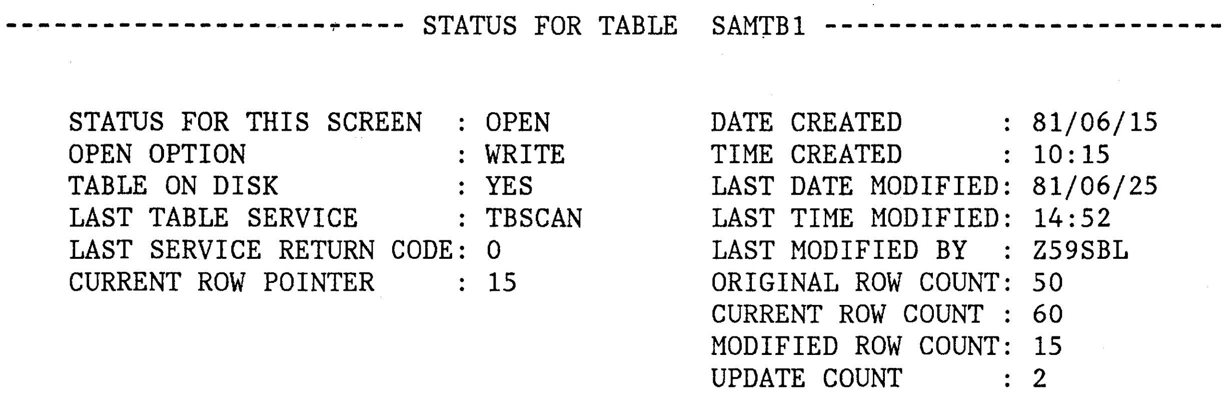
DATE CREATED Date the table was created

TIME CREATED Time the table was created

LAST DATE MODIFIED Date the table was last modified

LAST TIME MODIFIED Time the table was last modified

LAST MODIFIED BY User id of the user who last modified the table



```
------------------------- STATUS FOR TABLE  SAMTB1 ------------------------

   STATUS FOR THIS SCREEN  : OPEN          DATE CREATED      : 81/06/15
   OPEN OPTION             : WRITE         TIME CREATED      : 10:15
   TABLE ON DISK           : YES           LAST DATE MODIFIED: 81/06/25
   LAST TABLE SERVICE      : TBSCAN        LAST TIME MODIFIED: 14:52
   LAST SERVICE RETURN CODE: 0             LAST MODIFIED BY  : Z59SBL
   CURRENT ROW POINTER     : 15            ORIGINAL ROW COUNT: 50
                                           CURRENT ROW COUNT : 60
                                           MODIFIED ROW COUNT: 15
                                           UPDATE COUNT      : 2
```

Figure 102. Dialog Test - Table Status Display for an Open Table

ORIGINAL ROW COUNT The number of rows that were added to a newly created table before closing the table for the first time.

CURRENT ROW COUNT The number of rows currently in the table.

MODIFIED ROW COUNT The number of rows in the table that have been modified at least once. A row that has been added to an existing table is also considered a modified row.

UPDATE COUNT Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

Note: Tables option 4.3 (modify row) allows you to modify a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when modifying a key value.

BROWSE ISPF LOG (OPTION 7.5)

The browse log option allows you to display data recorded in the ISPF log. You can use all the browse commands while looking at the ISPF log.

Sometimes the log is not available for browsing. This can occur when:

- The log file is empty.

- The log file was not created for this session. (The primary page allocation has been set to 0 using ISPF/PDF option 0.2.)

- No data has been written to the log during this session, and although the log file exists and is not empty, the user did not pass through normal termination during the last ISPF session (for example, an ABEND may have occurred). The log may be browsed if an action is taken that causes a log entry to be written.

The following trace output is written to the ISPF log:

- Trace header entries

- Function trace entries

- Variable trace entries

Each type of entry follows the format of other log entries: a short summary on the left and a detailed entry on the right.

Trace Header

The first line of trace data is a trace header that identifies the trace and shows the current application ID, the current function, and the current screen as detailed description. For split screen mode, the original screen is 1 and the screen generated by the SPLIT command is 2. The summary section of the header entry identifies the entry as a dialog trace. The trace header entry is written during the test session whenever a function or variable trace entry is to be written for a different (from the last) application, function, or screen.

For example, a trace of logical screen 1 of function TESTF1 in application ISR would place the following line in the ISPF log:

```
DIALOG TRACE ---- - APPLICATION(ISR)  FUNCTION(TESTF1)  SCREEN(1)
```

Function Trace Entries

A pair of function trace entries, a BEGIN entry and an END entry, is generated during a function trace for each traced dialog service that is invoked. A service may be called from a user dialog that is currently executing, or from a dialog test action on behalf of the user. The summary portion of each of these entries shows the name of the dialog service, whether it is the beginning or the end of its execution, and whether it was invoked indirectly from a Dialog Test panel. (If the word TEST does not appear, the user's dialog invoked the service directly.) For END entries only, the service return code is shown on a second line. The detailed section of the log entries contains an image of the service call and the parameters used to invoke that service, using two lines if necessary; for example:

```
DISPLAY .. BEGIN ... TEST - DISPLAY PANEL(XYZ)
           •
           •
           •
DISPLAY .. END.. ... TEST - DISPLAY PANEL(XYZ)
  ..RETURN CODE (0)
```

There can be many log entries between the begin and end entries. For example, any active variable traces may cause log entries during a SELECT trace.

The following should be noted about the service call image:

- The image is truncated after the second line.
- ISPEXEC calls are shown as coded in the dialog.
- ISPLINK (ISPLNK) calls are displayed with their parameter values separated by commas. Name lists are shown as coded in the dialog, in string format or structure format. (Structure format includes the count, element length, and list of names.) For variable services parameters whose context is defined by the "name-list" parameter on the service call, the first four bytes of the parameter value are displayed in hexadecimal (X'nnnnnnnn').
- Dialog test calls are shown using the command call format without the ISPEXEC prefix.

Variable Trace Entries

Two variable trace entry lines are generated for each variable trace log entry. The variable may be referenced or set by a user dialog (directly or indirectly through the use of a dialog service), or explicitly set by a dialog test option on behalf of a user. The summary parts of these entry lines identify the trace. Line one shows the name of the variable, the pool in which it exists (F for function, S for shared, P for profile), and an indicator (TEST) if a dialog test option set the value. Line 2 shows the operation performed for the variable (GET, PUT, or CHG) and the name of the dialog service that performed the operation (for non-TEST entries). The current value of the variable is printed in the detail section of the log entry and can span two lines; for example:

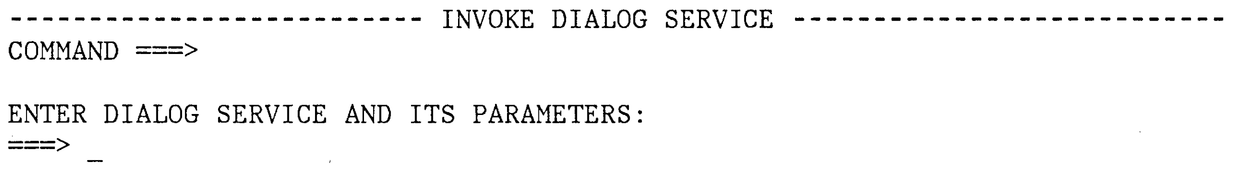
```
LIB1.... POOL(P) .... - VALUE(FLAG)
 ...GET by EDIT       -
```

Note: The value is truncated after the second line.

If the variable value contains nondisplayable characters, the value is displayed in hexadecimal format (X'nnnnnnnn').

INVOKE SERVICE (OPTION 7.6)

The dialog services option allows you to execute a dialog service by entering the service command invocation with or without the ISPEXEC characters. Figure 103 shows the invoke service panel.



```
------------------------- INVOKE DIALOG SERVICE --------------------------
COMMAND ===>

ENTER DIALOG SERVICE AND ITS PARAMETERS:
===> _
```

Figure 103. Dialog Test - Invoke Service Entry Panel

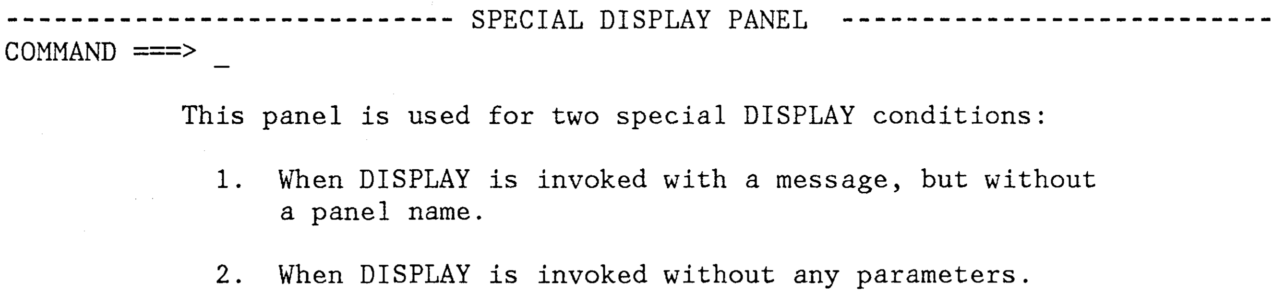
For example, if you wish to display panel XYZ, enter:

```
===> DISPLAY PANEL(XYZ)
           or
===> ISPEXEC DISPLAY PANEL(XYZ)
```

The service is invoked when you press the ENTER key. You are informed of the service's completion and return code.

You can call any dialog service valid in the command environment (except CONTROL) at a breakpoint or before invoking a function.

If you issue the DISPLAY service call with only a message parameter, or if you issue the DISPLAY service call with no parameter, a special display panel is shown (Figure 104).



```
---------------------------- SPECIAL DISPLAY PANEL ---------------------------
COMMAND ===> _

           This panel is used for two special DISPLAY conditions:

             1.  When DISPLAY is invoked with a message, but without
                 a panel name.

             2.  When DISPLAY is invoked without any parameters.
```

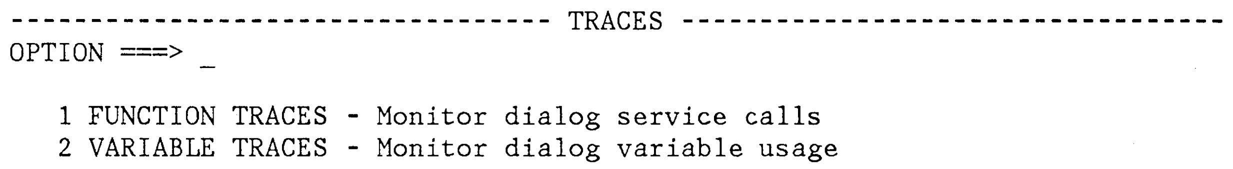
Figure 104. Dialog Test - Special Display Panel

TRACES (OPTION 7.7)

The traces option allows you to define, change, and delete trace specifications. You can trace executed dialog services, except for the VPUT service issued from a panel, and referenced dialog variables during dialog execution. Trace data is placed in the transaction log, where you can browse it using dialog test option 5, or print it when you exit from ISPF.

Since tracing may degrade dialog performance and create large amounts of output, care should be taken in setting the scope of trace definitions.

When you select this option, you are shown a selection panel (Figure 105) on which you can indicate the type of trace (function or variable) you wish to define. The trace capabilities are described below.



```
--------------------------------- TRACES ---------------------------------
OPTION ===> _

   1 FUNCTION TRACES - Monitor dialog service calls
   2 VARIABLE TRACES - Monitor dialog variable usage
```

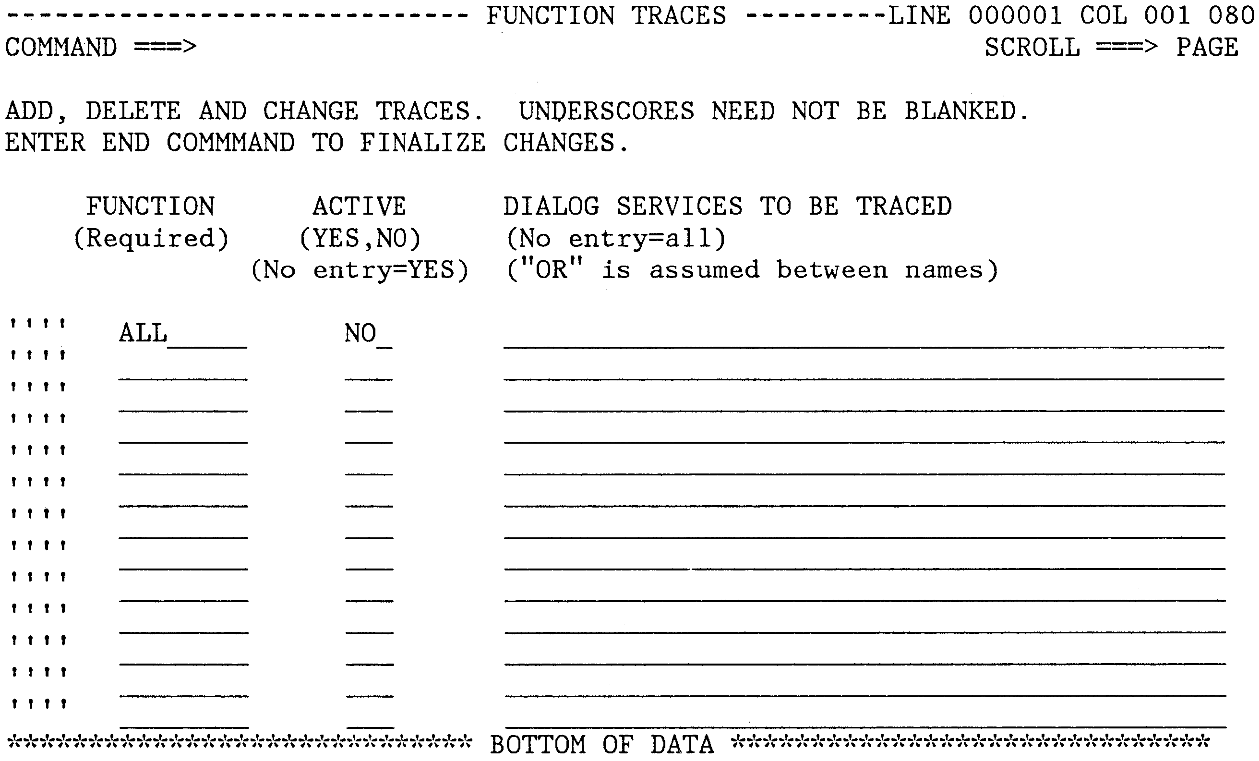
Figure 105. Dialog Test - Traces Selection Panel

Function Trace (Trace Option 1)

The function trace option is used to establish criteria for recording the names of dialog service calls, the service parameters, and return code in the ISPF log. Service calls made by the dialog or during test processing (e.g., using dialog test option 2 to display a panel) are recorded. Whenever a new application and/or function has data recorded, a header is placed in the trace.

When you select the function trace option, you are shown a scrollable panel (Figure 106) that lists all currently defined function traces.

You may add, delete, and modify function trace definitions using this panel before invoking a function or at a breakpoint.



```
----------------------------- FUNCTION TRACES ---------LINE 000001 COL 001 080
COMMAND ===>                                                 SCROLL ===> PAGE

ADD, DELETE AND CHANGE TRACES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMMAND TO FINALIZE CHANGES.

     FUNCTION      ACTIVE      DIALOG SERVICES TO BE TRACED
    (Required)    (YES,NO)     (No entry=all)
               (No entry=YES)  ("OR" is assumed between names)

''''   ALL_____      NO_       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
''''   ________      ___       _____________________________________________
       ________      ___       _____________________________________________
****************************** BOTTOM OF DATA ********************************
```

Figure 106. Dialog Test - Function Trace Entry Panel

Each line defines a function trace, showing:

'''' Line command area.

FUNCTION The name of the user function in which the trace should occur, or ALL to trace every dialog function. Initially, ALL is presented on the display but is not activated. Change the NO to a YES in the ACTIVE column to start such a trace. If you want to trace a function whose name is ALL, enclose that name in apostrophes ('ALL') to distinguish it.

ACTIVE Whether the trace is to be active now:

yes, y The trace is currently active.

no, n The trace is currently not active.

no entry The trace is currently active.

DIALOG SERVICES TO BE TRACED Names of dialog services to be traced. No entry in this field indicates all calls to dialog services for the function are to be traced.

All function traces exist until you leave dialog test, or until you delete them from this panel. Enter new information by overtyping the existing data. The underscores are pad characters to indicate the starting and ending positions for each field and need not be blanked out. You can create several function traces before you press the ENTER key.

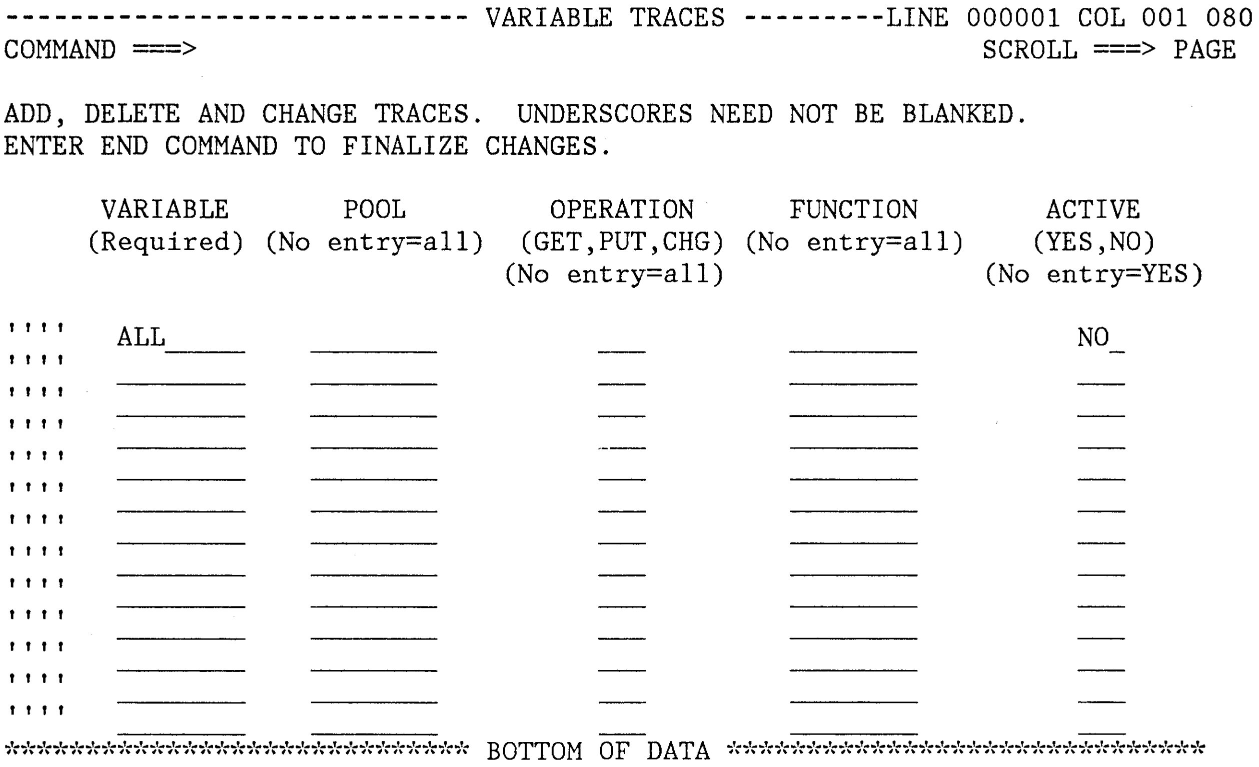
During dialog processing, to determine whether the criteria for a function trace have been met, dialog test executes a logical "AND" of the three fields (FUNCTION, ACTIVE, and DIALOG SERVICES) specified for that function trace. Within the DIALOG SERVICES field, a logical "OR" is performed to determine whether a particular dialog service has been matched. Therefore, if you want more than one trace for a function, you should create multiple rows.

The function trace option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. These commands are described in the section entitled "Dialog Test Commands."

Variable Trace (Trace Option 2)

The variable trace option is used to establish criteria for recording variable usage. A variable's usage is recorded if an ISPF service is directly asked to operate on the variable (e.g., VGET, VPUT, VCOPY), or if an ISPF service is indirectly asked to operate on the variable (e.g., DISPLAY). Variables changed under the variables option (dialog test option 3) are also recorded if the trace specifications are met.

When you select the variable trace option, you are shown a scrollable display (Figure 107) which lists all currently defined variable traces. You may add, delete, and modify variable trace definitions at a breakpoint, or by using this panel before invoking a function.



```
----------------------------- VARIABLE TRACES ---------LINE 000001 COL 001 080
COMMAND ===>                                                 SCROLL ===> PAGE

ADD, DELETE AND CHANGE TRACES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

      VARIABLE       POOL        OPERATION     FUNCTION        ACTIVE
     (Required) (No entry=all)  (GET,PUT,CHG) (No entry=all)   (YES,NO)
                               (No entry=all)               (No entry=YES)

''''   ALL_____    ________          ___         ________          NO_
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
''''   ________    ________          ___         ________          ___
****************************** BOTTOM OF DATA ******************************
```

Figure 107. Dialog Test - Variable Trace Entry Panel

Each line defines a variable trace, showing:

'''' Line command area

VARIABLE Name of the variable to be traced, or ALL to indicate tracing all variables. Initially, ALL is presented on the display but is not activated. Change the NO in the ACTIVE column to YES to start such a trace. If you wish to trace a variable whose name is ALL, enclose that name in apostrophes ('ALL') to distinguish it.

POOL Pool of interest for variable tracing:

f Function variable pool

s Shared variable pool

p Profile variable pool

no entry All pools

OPERATION Type of variable reference to trace:

- **get, g** Accesses to the variable's value
- **put, p** Stores to the variable's value
- **chg, c** Changes to the variable's value
- no entry All references to variable are traced

FUNCTION The user dialog function for which the trace is defined. If no entry, indicates this variable will be traced for all functions.

ACTIVE Indication of whether the trace is to be active:

- **yes, y** The trace is currently active.
- **no, n** The trace is currently not active.
- no entry The trace is currently active.

All variable trace definitions exist until you exit from dialog test, or until you delete them from this panel. Enter new information by overtyping the existing data. The underscores are pad characters to indicate the start and end of each field and need not be blanked out. You can create several variable traces before you press the ENTER key.

During dialog processing, to determine whether the criteria for a variable trace have been met, dialog test executes a logical "AND" of the fields (VARIABLE NAME, POOL, OPERATION, FUNCTION, and ACTIVE) specified for that variable trace. Therefore, if you want more than one trace for a variable, you should create multiple rows.

The variable trace option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. These commands are described in the section entitled "Dialog Test Commands."

BREAKPOINTS (OPTION 7.8)

A **breakpoint** is a location at which the execution of a user dialog is suspended so that dialog test facilities may be used. The breakpoints option permits you to indicate where such temporary suspensions should occur. At a breakpoint, control is given to the user, and user dialog data (tables, variables, etc.) may be examined and manipulated. In addition, new test conditions, such as traces and other breakpoints, may be specified.

Breakpoints are located immediately before a dialog service receives control or after it relinquishes control. Breakpoint definitions cause special handling within the ISPLINK (ISPLNK) and ISPEXEC interfaces to dialog services; no user dialog code is modified. When the criteria for a breakpoint are satisfied, the user's dialog is "suspended," and he is able to perform any of the functions provided by the dialog test breakpoint primary option menu. The VPUT service issued from a panel cannot be used as a breakpoint.

Defining Breakpoints (Breakpoints Panel)

When you select the breakpoints option, you are shown a scrollable display (Figure 108), showing all currently defined breakpoints for this session. You may add, delete, and modify breakpoint definitions using this panel before invoking a function or at a breakpoint.

Each line defines a breakpoint, and shows:

'''' Line command area

SERVICE Name of the dialog service at which to interrupt dialog execution. This field is required.

WHEN Indication of when the breakpoint should occur:

- **before, b** Before the service receives control
- **after, a** After the service completes execution
- **rnn** After the service completes execution, if the return code is the integer nn
- no entry Before and after service execution

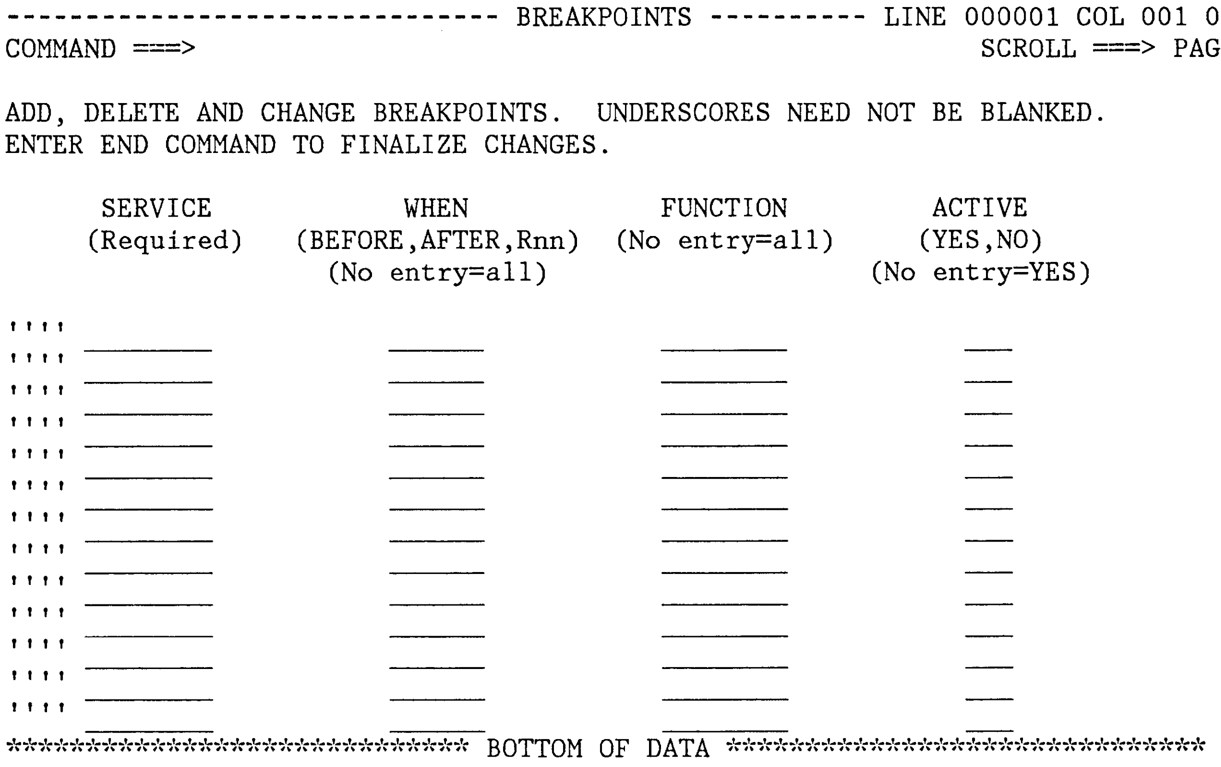
FUNCTION The user dialog function (program or command) that must be executing for the breakpoint to be taken. No entry in this field indicates that the breakpoint may occur for all functions.

ACTIVE Indication of whether the breakpoint is to be active at this time:

- **yes, y** It is currently active
- **no, n** It is currently not active
- no entry It is currently active

QUAL If present at the end of a row, indicates that qualification data exists for the breakpoint. This field is nonmodifiable. For additional information see "Qualifying a Breakpoint," below.

Empty lines (all input fields contain underscores) are added to the initial display to fill up the screen. If you delete all the lines used for defining breakpoints, the display is automatically refreshed with enough empty lines to fill the screen again.



```
------------------------------ BREAKPOINTS ---------- LINE 000001 COL 001 080
COMMAND ===>                                              SCROLL ===> PAGE

ADD, DELETE AND CHANGE BREAKPOINTS.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

      SERVICE           WHEN            FUNCTION        ACTIVE
     (Required)  (BEFORE,AFTER,Rnn)  (No entry=all)     (YES,NO)
                   (No entry=all)                     (No entry=YES)

''''
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
'''' ________        ______           ________          ___
****************************** BOTTOM OF DATA ******************************
```

Figure 108. Dialog Test - Breakpoints Entry Panel

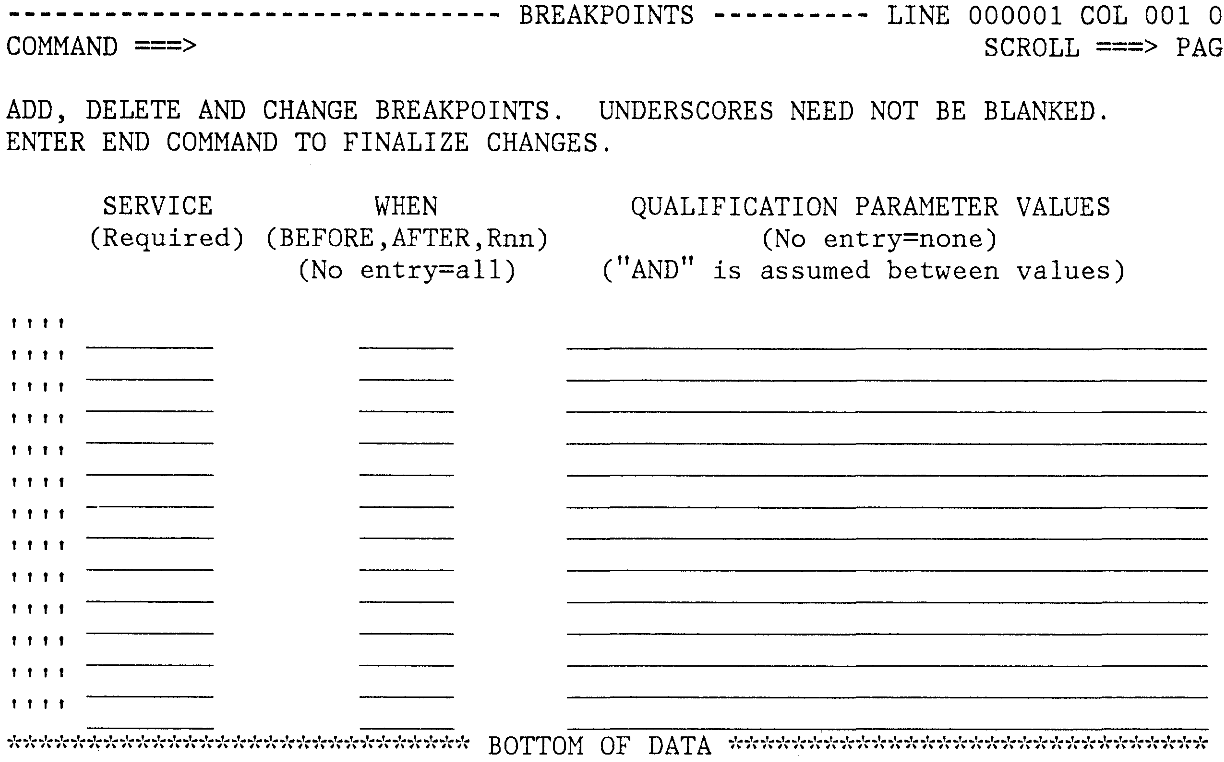
All breakpoints exist until you end or cancel your dialog test session, or until you delete them from this panel. Enter new information by overtyping the existing data. The underscores are pad characters to indicate the starting and ending positions for each field and need not be blanked out. You can create several breakpoints before you press ENTER.

From the breakpoint entry panel, you can use the CANCEL, END, LOCATE, and QUAL dialog test primary commands, and the D, I, and R dialog test line commands. These commands are described in the section entitled "Dialog Test Commands."

Qualifying a Breakpoint (Qualification Panel)

The qualification panel allows you to further constrain the conditions under which a breakpoint is to occur. You may list parameter data with which the named service must have been invoked.

The qualification panel is displayed when you enter the QUAL primary command on the breakpoints panel. The FUNCTION and ACTIVE columns are overlaid with a column of data, headed with QUALIFICATION PARAMETER DATA, which was logically off the screen to the right of the breakpoints panel display (Figure 109). The RESUME primary command is used to resume the format of the breakpoints panel.



```
------------------------------ BREAKPOINTS ---------- LINE 000001 COL 001 080
COMMAND ===>                                                 SCROLL ===> PAGE

ADD, DELETE AND CHANGE BREAKPOINTS.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

      SERVICE           WHEN           QUALIFICATION PARAMETER VALUES
     (Required) (BEFORE,AFTER,Rnn)            (No entry=none)
                  (No entry=all)     ("AND" is assumed between values)

''''
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
'''' ________        ______      ________________________________________
     ________        ______      ________________________________________
****************************** BOTTOM OF DATA ******************************
```

Figure 109. Dialog Test - Qualification Panel

The qualification panel lines correspond to the lines on the breakpoints panel; the **SERVICE** and **WHEN** fields are described previously. In the **QUALIFICATION PARAMETER VALUES** field, you can enter, for all services except SELECT, any combination of the following:

- One or more parameter values (separated by blanks) that the dialog passes to the service. No order is implied by the specification of the parameter values.

  For example, if you want a breakpoint to occur when message ABC0001 is coded on a DISPLAY service request, you should specify ABC001. If the breakpoint should occur only when message ABC0001 and panel XYZ are both coded, you should specify ABC0001 XYZ.

- One or more command call keywords (separated by blanks) that have nonblank values when a dialog calls the service. For ISPLINK (ISPLNK) calls, the keywords matching the calling sequence parameter positions are used.

  For example, if you want a breakpoint to occur whenever the DISPLAY service is invoked with a message, then specify MSG.

For the SELECT service, you can enter one or more parameter strings that would be entered on the SELECT service call. A string is defined as a series of characters that is delimited by a blank, a comma, a quote, or a left or right parenthesis.

For example, if a SELECT call is SELECT PGM(ABC) PARM(1 2 3 5 '6'), then any or all of the following strings can be used: SELECT, PGM, ABC, 1, 2, 3, 5, 6.

For a breakpoint to be taken, all qualification data listed must be matched.

All line commands and modification capabilities are still available on the qualification panel.

During dialog processing, to determine whether the criteria for a breakpoint have been met, dialog test executes a logical "AND" of all fields (SERVICE, WHEN, FUNCTION, ACTIVE, and QUALIFICATION) specified for that breakpoint. Therefore, if you want more than one breakpoint for an ISPF service, you should create multiple rows.

When you use the breakpoints option, be aware of the following items:

QUALIFICATION If you plan to qualify several breakpoints, it may be more efficient to specify all breakpoint data on the qualification panel.

END COMMAND You can use the END primary command from either the breakpoints panel or the breakpoints qualification panel.

INPUT ERRORS You must correct input errors before leaving any display (using the END, QUAL, or RESUME command). You may use the CANCEL command to terminate the breakpoints option, even if input errors remain on the display.

SYNTAX CHECKING A dialog service call must pass a basic syntax check before a breakpoint will be honored.

CONTROL DISPLAY If you encounter a breakpoint during execution, any settings of CONTROL DISPLAY LINE or SM in your dialog are lost.

TUTORIAL (OPTION 7.T)

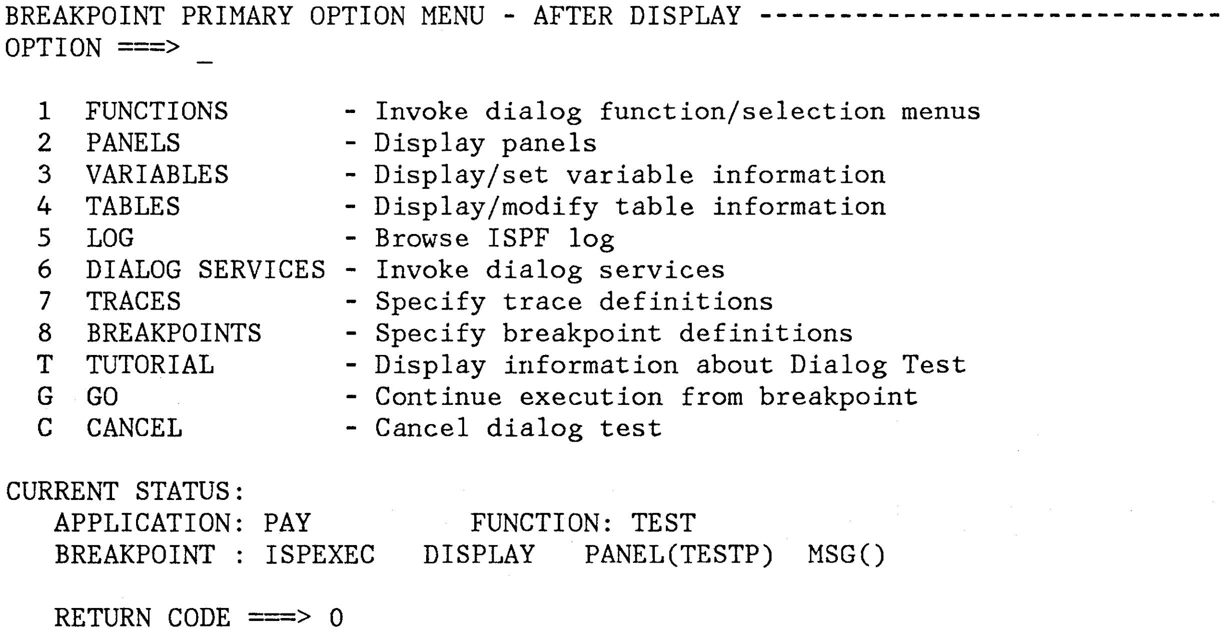
The tutorial option allows you to display information about the dialog test facilities.

EXIT (OPTION 7.X)

The exit option terminates your dialog test session. All trace and breakpoint definitions are lost.

ENCOUNTERING A BREAKPOINT

If you invoke a dialog function or selection panel and encounter a breakpoint, the dialog test breakpoint primary option menu is displayed. Figure 110 shows this selection panel at a breakpoint just after invoking the ISPF DISPLAY service while executing the TEST function in application PAY.



```
BREAKPOINT PRIMARY OPTION MENU - AFTER DISPLAY ----------------------------
OPTION ===> _

   1  FUNCTIONS       - Invoke dialog function/selection menus
   2  PANELS          - Display panels
   3  VARIABLES       - Display/set variable information
   4  TABLES          - Display/modify table information
   5  LOG             - Browse ISPF log
   6  DIALOG SERVICES - Invoke dialog services
   7  TRACES          - Specify trace definitions
   8  BREAKPOINTS     - Specify breakpoint definitions
   T  TUTORIAL        - Display information about Dialog Test
   G  GO              - Continue execution from breakpoint
   C  CANCEL          - Cancel dialog test

CURRENT STATUS:
   APPLICATION: PAY          FUNCTION: TEST
   BREAKPOINT : ISPEXEC   DISPLAY   PANEL(TESTP)  MSG()

   RETURN CODE ===> 0
```

Figure 110. Dialog Test - Breakpoint Primary Option Menu

Like the dialog test primary option menu, the breakpoint primary option menu allows you to use the RETURN command from any one of the selected test options to process a redisplay of the breakpoint primary option menu. At the breakpoint primary option menu, the END and RETURN commands have no effect. You must use the GO option to terminate processing at this breakpoint and continue executing the dialog being tested, or the CANCEL option to cancel the dialog test option. This protects against inadvertent loss of data.

The breakpoint primary option menu contains all of the options of the dialog test primary option menu, and as such, presents all of the dialog test functions to you. It also displays the following information:

Panel Title BREAKPOINT PRIMARY OPTION MENU

BEFORE or AFTER An indication of whether the dialog has been suspended BEFORE or AFTER the service has executed.

Service Name The name of the service at which the dialog has been suspended. In Figure 110, the service name is **DISPLAY**.

CURRENT STATUS:

APPLICATION The application identifier of the suspended user dialog.

FUNCTION The program or command name of the suspended user dialog.

BREAKPOINT Up to two display lines showing an image of the dialog service call. Truncation occurs after the second line.

ISPEXEC calls are shown as coded.

ISPLINK (ISPLNK) calls are displayed with their parameter values separated by commas. Name lists are shown as coded in the dialog, in string format or in structure format (structure format includes the count, element length, and list of names). For variable services parameters whose context is defined by the "name-list" parameter on the service call (for example, the variable value areas for a VDEFINE), the first four bytes of the parameter value are displayed in hexadecimal (X'nnnnnnnn').

RETURN CODE The dialog service return code. This field is displayed only if the breakpoint occurs **AFTER** the dialog service has executed. The return code field is modifiable; its value is passed back to the dialog (as the service's) when you select the GO option. This facilitates the testing of dialog error handling.

GO

The GO option continues dialog execution from a breakpoint. The user dialog resumes processing from the point at which it was suspended.

CANCEL

The CANCEL option terminates dialog testing and redisplays the first primary option menu you displayed at the beginning of your ISPF session. All trace and breakpoint definitions are lost when dialog test is terminated.

When a user dialog encounters a breakpoint, the current dialog environment is saved. When you select the GO option, the environment is restored, except for the following:

- If you change variable, table, and file tailoring data at a breakpoint, these actions are performed as an extension of the suspended dialog; it is as though the dialog had taken all the actions itself during execution.
- If you modify the service return code (on the breakpoint primary option menu), the new return code is passed back to the dialog (as though the service had set the new code itself).
- If you execute the PANELID command at the breakpoint, the last setting for displaying panel identifiers is retained.
- If any CONTROL service settings for DISPLAY LINE or DISPLAY SM were in effect before the breakpoint, such settings are lost.

Note that the manipulation of one dialog part may cause a change to another dialog part. For example, if a panel is displayed, variables may be set.

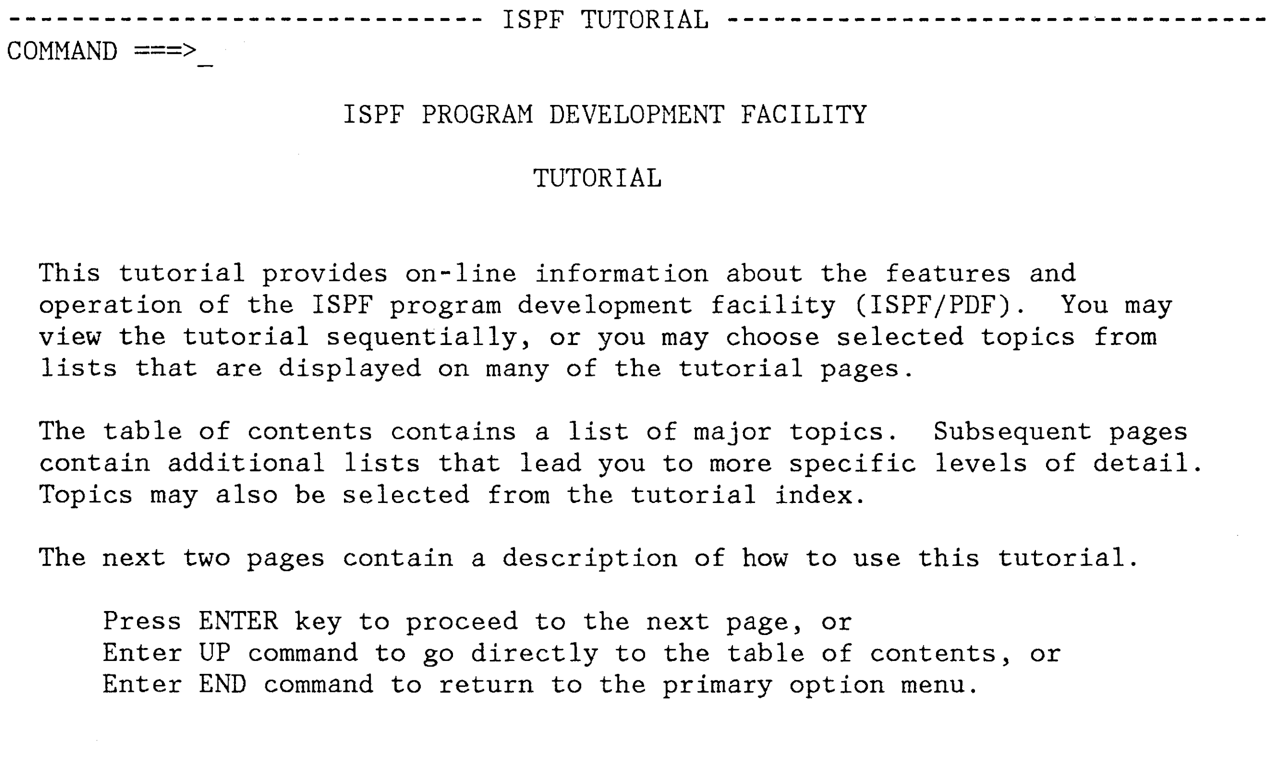
All trace and breakpoint definitions are lost if you select the CANCEL option from the breakpoint primary option menu.

CHAPTER 12. TUTORIAL (OPTION T)

The tutorial option provides immediate online reference and instruction on how to use PDF. You can invoke it by selecting option T from the primary option menu or by using the HELP command. You can scan the tutorial sequentially from beginning to end, or you can select specific topics from an alphabetic index or table of contents.

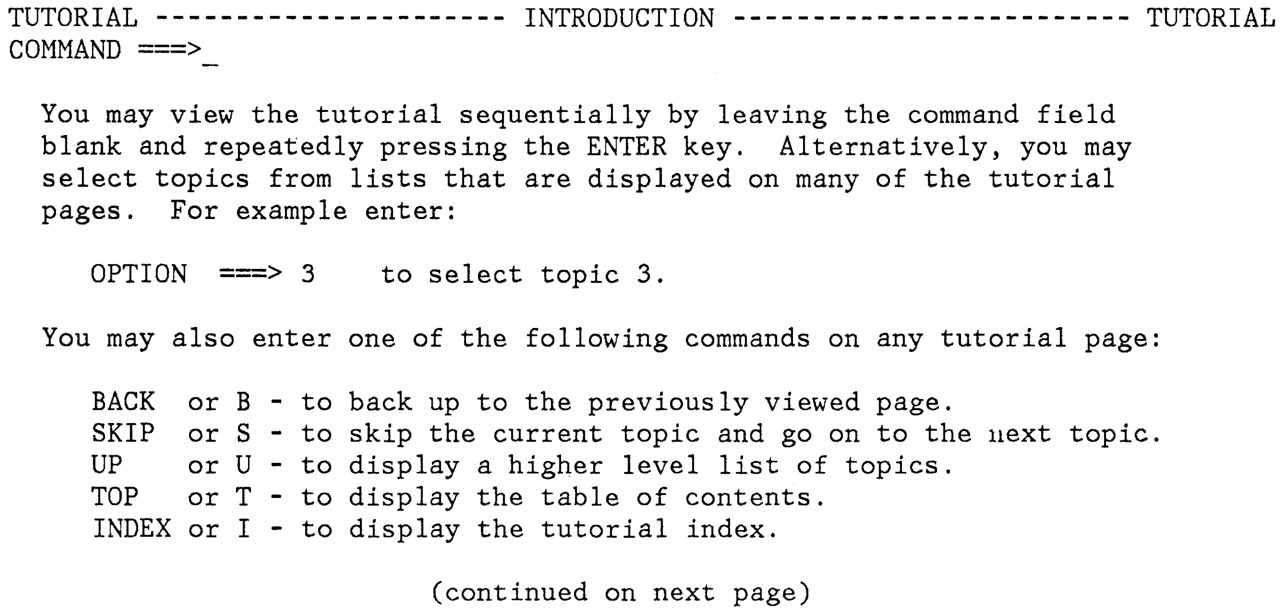
When you invoke the tutorial from the primary option menu, introductory pages are displayed to explain how the tutorial is organized and how it might be used (Figure 111). Next, a table of contents is displayed from which you may select a topic by entering the desired section number.

When you invoke the tutorial via the HELP command, the appropriate section of the tutorial is entered based on what you were doing when you requested help.



```
----------------------------- ISPF TUTORIAL ----------------------------------
COMMAND ===>_

                    ISPF PROGRAM DEVELOPMENT FACILITY

                                TUTORIAL

  This tutorial provides on-line information about the features and
  operation of the ISPF program development facility (ISPF/PDF).  You may
  view the tutorial sequentially, or you may choose selected topics from
  lists that are displayed on many of the tutorial pages.

  The table of contents contains a list of major topics.  Subsequent pages
  contain additional lists that 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 key to proceed to the next page, or
      Enter UP command to go directly to the table of contents, or
      Enter END command to return to the primary option menu.
```

Figure 111 (Part 1 of 2). Tutorial - First Two Pages



```
TUTORIAL --------------------- INTRODUCTION ------------------------ TUTORIAL
COMMAND ===>_

  You may view the tutorial sequentially by leaving the command field
  blank and repeatedly pressing the ENTER key.  Alternatively, you may
  select topics from lists that are displayed on many of the tutorial
  pages.  For example enter:

     OPTION  ===> 3    to select topic 3.

  You may also enter one of the following commands 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 111 (Part 2 of 2). Tutorial - First Two Pages

When viewing the tutorial, select topics by number, or simply press the ENTER key to view the next topic. On any panel, you may also enter the following commands:

```
BACK  or B - return to the previously viewed page
SKIP  or S - skip to the next topic
UP    or U - display a higher level list of topics
TOP   or T - display the table of contents
INDEX or I - display the index
```

When you use the tutorial, the scrolling commands are interpreted as follows:

```
UP            - display higher level list of topics
DOWN  (skip)  - skip to the next topic
LEFT  (back)  - display the previous tutorial page
RIGHT (next)  - display the next tutorial page
```

The ENTER key also means "next."

While viewing the tutorial, you can enter the HELP command at any time to display a one-page summary of how to use the tutorial.

To terminate the tutorial function, enter either the END or RETURN command, which causes a return either to the primary option menu or to the display from which you requested help.

APPENDIX A. USAGE SCENARIO

This usage scenario is an example of PDF terminal operations. It is intended for new users as a quick introduction to PDF. It may also be useful as a demonstration that PDF has been properly installed and is operational.

The scenario requires the installation of file ISRSAMP ASSEMBLE (included on the PDF basic distribution tape), which should be available for read-only access on the PDF system minidisk.

Prior to starting this sample problem, you should make sure that there are no files in your virtual reader.

ACTION	RESULT
1. LOGON TO VM/SP AND IPL CMS	1. READY MESSAGE
2. KEY "PDF"	2. ISPF/PDF PRIMARY OPTION MENU
3. SELECT UTILITIES (OPTION 3)	3. UTILITY SELECTION PANEL
4. SELECT FILE UTILITY (OPTION 2)	4. FILE UTILITY PANEL
5. SELECT OPTION S (SPECIFY) AND ENTER: PROJECT : SAMPLE LIBRARY : USERID (YOUR VM USERID) TYPE : ASM	5. SPECIFY NEW LIBRARY PANEL
6. ENTER THE FOLLOWING: ORGANIZATION : S RECORD FORMAT : F RECORD LENGTH : 80 LINK ACCESS MODE : W (LEAVE THE OTHER FIELDS BLANK)	6. 'SAMPLE.USERID.ASM' IS SPECIFIED AS AN ISPF LIBRARY. FILE UTILITY PANEL IS DISPLAYED.
7. SELECT OPTION S AGAIN AND ENTER: PROJECT : SAMPLE LIBRARY : USERID (YOUR VM USERID) TYPE : TEXT	7. SPECIFY NEW LIBRARY PANEL
8. LEAVE EVERYTHING THE SAME. JUST PRESS ENTER.	8. 'SAMPLE.USERID.TEXT' IS SPECIFIED AS AN ISPF LIBRARY. FILE UTILITY PANEL IS DISPLAYED.
9. PRESS PF3	9. UTILITY SELECTION PANEL
10. PRESS PF3	10. ISPF/PDF PRIMARY OPTION MENU
11. SELECT MOVE/COPY (OPTION 3.3)	11. MOVE/COPY UTILITY PANEL
12. SELECT OPTION C (COPY) AND ENTER CMS FILE NAME: ISRSAMP ASSEMBLE	12. SECOND MOVE/COPY PANEL

ACTION	RESULT
13. BLANK OUT CMS FILE ID AND ENTER: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ASMTEST	13. ISRSAMP ASSEMBLE FILE IS COPIED TO YOUR LIBRARY (SAMPLE.USERID.ASM) AS MEMBER ASMTEST. THE FIRST MOVE/COPY PANEL IS REDISPLAYED.
14. PRESS PF3	14. ISPF/PDF PRIMARY OPTION MENU
15. SELECT BROWSE (OPTION 1)	15. BROWSE ENTRY PANEL
16. BLANK OUT CMS FILE ID AND ENTER: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : (LEAVE BLANK)	16. BROWSE MEMBER 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 TEXT 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: ===> FIND COMMENT	20. STRING "COMMENT" IS INTENSIFIED.
21. FIND NEXT OCCURRENCE OF "COMMENT" BY PRESSING PF5	21. SECOND OCCURRENCE OF "COMMENT" IS INTENSIFIED.
22. PRESS PF3	22. MEMBER LIST
23. PRESS PF3	23. BROWSE ENTRY PANEL
24. PRESS PF3	24. ISPF/PDF PRIMARY OPTION MENU
25. SELECT EDIT (OPTION 2)	25. EDIT ENTRY PANEL
26. SPECIFY ISPF LIBRARY: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ASMTEST	26. EDIT DATA DISPLAY OF SAMPLE.USERID.ASM(ASMTEST).

ACTION	RESULT
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 ===> AUTOLIST ON	30. THIS SETS AUTOLIST 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 ISPF LIST FILE. THE EDIT ENTRY PANEL IS DISPLAYED.
33. PRESS PF3	33. ISPF/PDF PRIMARY OPTION MENU
34. SELECT FOREGROUND (OPTION 4)	34. FOREGROUND SELECTION PANEL
35. SELECT SYSTEM ASSEMBLER	35. FOREGROUND ASSEMBLY PANEL
36. ENTER ASSEMBLER INPUT: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ISRASM LIST ID : ASMTEST1 ASSEMBLER OPTIONS : LIST TEST RENT	36. ASSEMBLER PROMPTER IS INVOKED. TERMINAL OUTPUT IS WRITTEN AT THE TOP OF A BLANK SCREEN. WHEN "MORE..." APPEARS, PRESS ENTER TO CONTINUE. (THIS ASSEMBLY MAY TAKE A FEW MINUTES.) THE ASSEMBLY LISTING IS DISPLAYED IN BROWSE MODE.

ACTION	RESULT
37. BROWSE THE LISTING, IF DESIRED, USING THE SCROLL COMMANDS. THEN PRESS PF3.	37. FOREGROUND PRINT PANEL IS DISPLAYED.
38. ENTER OPTION K (KEEP).	38. FOREGROUND ASSEMBLY PANEL IS DISPLAYED.
39. PRESS PF3	39. FOREGROUND SELECTION PANEL
40. PRESS PF3	40. ISPF/PDF PRIMARY OPTION MENU
41. SELECT BATCH (OPTION 5)	41. BATCH SELECTION PANEL
42. SELECT SYSTEM ASSEMBLER AND ENTER JOB INFORMATION AS REQUIRED BY YOUR INSTALLATION.	42. BATCH ASSEMBLY PANEL
43. ENTER ASSEMBLER INPUT: PROJECT : SAMPLE LIBRARY : USERID TYPE : ASM MEMBER : ISRASM LISTID : ASMTEST2 ASSEMBLER OPTIONS : OBJ,NODECK,LIST,TERM	43. CONTROL COMMANDS FOR ASSEMBLING ASMTEST ARE GENERATED. THE BATCH SELECTION PANEL IS DISPLAYED.
44. PRESS PF3	44. THE JOB IS SUBMITTED TO THE CMS BATCH MACHINE. WHEN "HOLDING" APPEARS, PRESS PA2. THE ISPF/PDF PRIMARY OPTION MENU IS DISPLAYED.

AS THE JOB EXECUTES IN THE BATCH MACHINE, STATUS MESSAGES WILL BE RETURNED TO YOU. IN PARTICULAR, WHEN THE ASSEMBLY IS COMPLETE, YOU WILL RECEIVE A MESSAGE INDICATING THAT THE TEXT AND LISTING FILES HAVE BEEN SPOOLED TO YOUR READER. DO NOT EXECUTE STEP 46 OF THIS SCENARIO UNLESS AND UNTIL YOU HAVE RECEIVED THE COMPLETION MESSAGE.

ACTION	RESULT
45. SELECT TUTORIAL (OPTION T)	45. START OF ISPF/PDF TUTORIAL IS DISPLAYED. FOLLOW THE DIRECTIONS TO LEARN MORE ABOUT ISPF/PDF. WHEN YOU ARE DONE, PRESS PF3 TO RETURN TO THE ISPF/PDF PRIMARY OPTION MENU.
46. SELECT READER UTILITY (OPTION 3.7)	46. READER UTILITY PANEL
47. SELECT INSERT (OPTION I) AND ENSURE THAT THE TARGET ISPF LIBRARY FOR THE TEXT FILE IS SAMPLE.USERID.TEXT	47. READER UTILITY PANEL IS REDISPLAYED, SHOWING THAT THE FILE WAS MOVED INTO YOUR ISPF LIBRARY, AND THAT THE LISTING IS NOW THE NEXT FILE IN YOUR READER.
48. SELECT READ (OPTION R) AND RENAME TO : ASMTEST2 LISTING THEN PRESS ENTER.	48. READER UTILITY PANEL IS REDISPLAYED, SHOWING THAT THE LISTING WAS LOADED ONTO YOUR A-DISK, AND THAT YOUR READER IS NOW EMPTY.
49. PRESS PF3	49. ISPF/PDF PRIMARY OPTION MENU

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 ISPF/PDF OPTIONS. WHEN YOU ARE DONE, RETURN TO THE PRIMARY OPTION MENU.

ACTION	RESULT
50. (AT PRIMARY OPTION MENU) PRESS PF3	50. ISPF TERMINATION PANEL
51. SELECT OPTION P FOR BOTH THE LOG AND LIST FILES, AND PRESS ENTER.	51. THE LOG AND LIST FILES ARE CLOSED AND RELEASED FOR PRINTING.
52. YOU ARE NOW OUT OF ISPF/PDF. TO EXIT FROM CMS, TYPE "LOGOFF".	52. END OF USAGE SCENARIO

APPENDIX B. ISPF LIBRARY SPECIFICATION AND USAGE

Before reading this appendix, please review the section entitled "ISPF Libraries" in Chapter 3.

GUIDELINES FOR LIBRARY SPECIFICATION

Each ISPF library must be specified with the ISPF/PDF file utility (option 3.2) before it can be used. The name of the library along with the following information must be specified:

ISPF library attributes:	
Organization	(S = set of files, M = MACLIB, T = TXTLIB)
Record format	(F = fixed, V = variable)
Record length	(1 to 32,767 bytes)
ISPF library location:	
Owner's id	(blank for your user id)
Device address	(blank for '191')
Link access mode for update:	
Mode letters	(M, W, or MW) (See "Link Access Modes")
For organization S (set of files):	
Filetype	(blank for ISPF-generated name)
For organization M or T (MACLIB or TXTLIB):	
Filename	(blank for ISPF-generated name)

Naming Conventions

An ISPF library is always identified by a 3-part 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 ISPF type qualifier need not conform to the standard CMS naming conventions, with one exception: If the library is to contain TEXT (object) modules, the ISPF 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 is maintained as a set of CMS sequential files, all with the same filetype. The CMS filenames are the same as the ISPF member names, and the CMS filetype may be anything that uniquely identifies the set of files on the minidisk.

M The library is maintained as a CMS file with a filetype of MACLIB. The CMS filename may be anything that uniquely identifies the MACLIB on the minidisk. The member names in the MACLIB are the same as the ISPF member names.

T The library is maintained as a CMS file with a filetype of TXTLIB. The CMS filename may be anything that uniquely identifies the TXTLIB on the minidisk. The member names in the TXTLIB are normally the same as the ISPF member names.

Note: TXTLIBs are maintained by PDF using the CMS TXTLIB command with the exception of the rename function of the library utility (option 3.1). Therefore, TXTLIB contents and member names are generated according to the CMS TXTLIB command specifications.

Libraries containing text to be processed by the SCRIPT/VS foreground option (option 4.9) must have an organization of "S", but they need not have an ISPF 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 ISPF 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 ISPF type qualifier, nor the CMS filetype (for organization "S").

Following are additional hints for specifying library organization:

- Libraries that 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:

  - The library has many members.

- The library is infrequently updated.

- The library contains members that are included or copied by other members during a compile or assembly. See "Foreground/Batch Processing," below, for a discussion of included members.

Record Format and Length

Libraries that have an organization of "M" or "T" must have a record format of "F" (fixed) and a record length of 80. Libraries that have an organization of "S" may have a record format of either "F" or "V" (variable) and a record length from 1 to 32,767. Note, however, that the ISPF/PDF editor does not process any data with a record length less than 10 or greater than 255.

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 ISPF library must be wholly contained on one minidisk, which is specified by the owner's user id and virtual device address.

More than one ISPF library may be assigned to the same minidisk, and ISPF libraries may co-reside on the same minidisk with other CMS files (which are not ISPF libraries). Be careful, however, to guard against conflicting CMS filetypes or filenames (see "Unique Identifier" and "Foreground/Batch Processing," below).

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 that 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 that 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 that are owned by the person who will be updating them, and that 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 that all have the same read password. The reason for this is that ISPF panels provide a single read password that 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 is successful only if no other user is currently accessing the minidisk (in either read or write mode).

M Link is successful if no other user is currently accessing the minidisk in write mode.

MW Link is successful in all cases.

If the library is to reside on a user's A-disk (which ISPF assumes has a virtual device address of 191), do not specify "MW" for the link access mode. ISPF does not protect against multiple write accessing to virtual device 191.

> **Note:** Generally, 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 ISPF libraries that 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 ISPF libraries or any other files) is through ISPF. If some accesses to the minidisk are done outside ISPF, specify a link access mode of "M" or "W".

ISPF provides appropriate protection to guard against destructive contention in multiple read/write accessing of ISPF 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 ISPF library. The recommended procedure is to let ISPF assign a unique identifier by leaving the filetype and filename fields blank on the "specify" panel. The ISPF-assigned identifier is of the form:

```
SPFnnnnc
```

where "nnnn" is a 4-digit number, and "c" is S, M, or T.

Keep in mind the following points:

- For an ISPF library, the CMS filetype (organization "S") or filename (organization "M" or "T") need not conform to the CMS naming convention.

- Since the ISPF library can be referenced by project, library, and type on any panel, the CMS filetype or filename is generally transparent to the user.

If a filetype or filename is entered on the "specify" panel, 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 are automatically included in the ISPF library. This may or may not be desired. In any case, the ISPF/PDF file utility displays a warning on the "confirm specify" panel that existing files are being included in the ISPF library.

PROCESSING CONSIDERATIONS

ISPF maintains information on all ISPF libraries specified to the VM system. Whenever a user enters project, library, and type on a panel, ISPF 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, ISPF 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). Link access mode "RR" allows the LINK to occur even if another user has the disk accessed in read or write status.

For the ACCESS command, ISPF 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 ISPF library is automatically detached (and the mode letter is released) when the user finishes the selected ISPF function.

Under certain circumstances, the automatic LINK and ACCESS can cause a temporary alteration to the user's normal search sequence for CMS files.

For example, suppose the user has explicitly accessed two minidisks, using mode letters A and C. Subsequently, the user specifies an ISPF library on some panel (such as browse). ISPF 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 panel. If the specified file exists on both the B and C disks (but not on A), the B copy is found first. The user may not be expecting this, since he did not explicitly access the B-disk.

Foreground/Batch Processing

The ISPF/PDF foreground and batch processing options (options 4 and 5) automatically perform the following operations if an ISPF 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 that is acceptable to the particular processing program (e.g., ASSEMBLE, COBOL, PLI). The CMS filename for the temporary file is the same as the ISPF member name.

- For any libraries in the ISPF concatenation sequence (including the first, or only, library) that are not organization "M", temporary MACLIBs are created and all members are copied from each ISPF 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 reside on the the user's A-disk. For batch, they 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 ISPF 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 ISPF TEXT library. For batch processing, the user may invoke the ISPF/PDF reader utility (option 3.7) to cause the TEXT module to be inserted into the ISPF TEXT library.

ISPF Library Statistics

Whenever an ISPF 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 that is the same as the unique identifier of the ISPF library.

Generally, the SPFSTATS file is transparent to the user, since the statistics are automatically created and maintained by the ISPF/PDF editor (if STATS mode is on), and automatically displayed on member 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 ISPF library members will remain intact.

If members of the ISPF library are modified outside of ISPF/PDF (with some other editor) or are specified on an ISPF/PDF panel using a CMS file id (rather than project, library, and type), the statistics are not properly updated, but the information in the member is still acceptable to ISPF.

> **Note:** If a user deletes all lines in a member of an ISPF library (using the PDF editor) and then saves the member, the statistics 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 is not 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. PDF LISTING FORMATS

If autolist mode is on, the PDF editor automatically generates a source listing upon completion of editing. Source listings may also be obtained via the Print option of the PDF library utility (option 3.1) and the move/copy utility (option 3.3).

A sample source listing is shown in Figure 112.

For an ISPF 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 nonblank character in each line. For ISPF library members with statistics, asterisks may or may not 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 nonzero, but differs from the current modification level of the member, a single asterisk (*) is printed.

- If the modification flag is nonzero, and has the same value as the current modification level of the member, two asterisks (**) are printed.

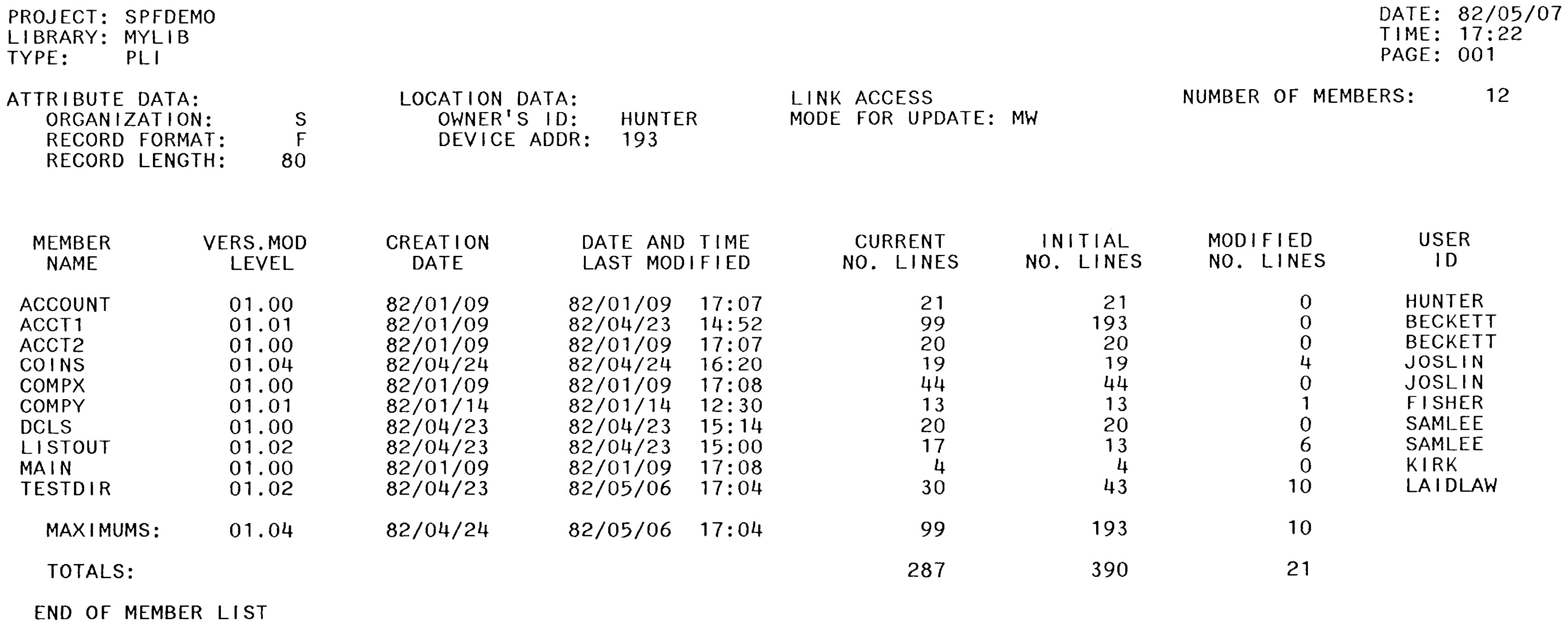
The asterisks allow the listing to be scanned quickly for lines that were added or changed since the version was created (*) and for lines that were added or changed during the last update (**).

PDF provides index listings, upon user request, via the "index" option of the library utility (option 3.1). A sample index listing for a ISPF library is shown in Figure 113. 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 ISPF log listing is shown in Figure 114. 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.

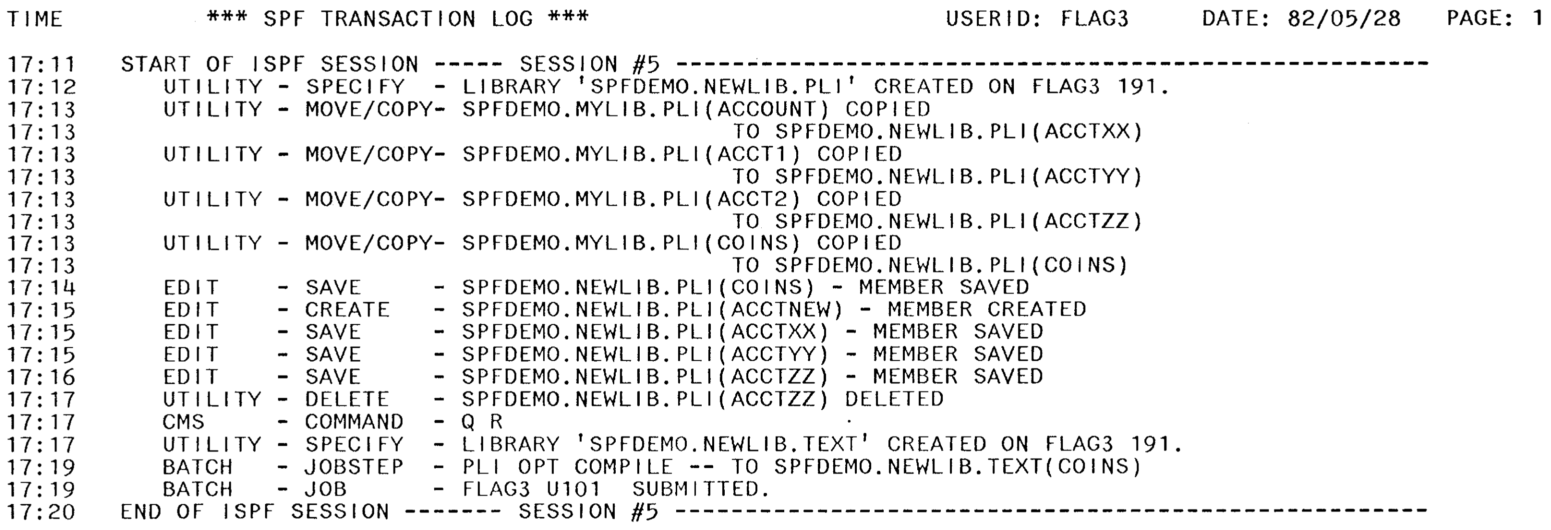
```
          PROJECT: SPFDEMO              MEMBER: COINS               DATE: 82/05/18                       COINS
          LIBRARY: MYLIB                LEVEL:  01.04               TIME: 17:22
          TYPE:    PLI                  USERID: HUNTER              PAGE: 01 OF 01
    START                                                                          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 112. Sample Source Listing



```
PROJECT: SPFDEMO                                                                                              DATE: 82/05/07
LIBRARY: MYLIB                                                                                                TIME: 17:22
TYPE:    PLI                                                                                                  PAGE: 001

ATTRIBUTE DATA:                LOCATION DATA:                 LINK ACCESS                 NUMBER OF MEMBERS:     12
   ORGANIZATION:      S           OWNER'S ID:   HUNTER        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         82/01/09     82/01/09  17:07            21            21             0         HUNTER
 ACCT1          01.01         82/01/09     82/04/23  14:52            99           193             0         BECKETT
 ACCT2          01.00         82/01/09     82/01/09  17:07            20            20             0         BECKETT
 COINS          01.04         82/04/24     82/04/24  16:20            19            19             4         JOSLIN
 COMPX          01.00         82/01/09     82/01/09  17:08            44            44             0         JOSLIN
 COMPY          01.01         82/01/14     82/01/14  12:30            13            13             1         FISHER
 DCLS           01.00         82/04/23     82/04/23  15:14            20            20             0         SAMLEE
 LISTOUT        01.02         82/04/23     82/04/23  15:00            17            13             6         SAMLEE
 MAIN           01.00         82/01/09     82/01/09  17:08             4             4             0         KIRK
 TESTDIR        01.02         82/04/23     82/05/06  17:04            30            43            10         LAIDLAW

   MAXIMUMS:    01.04         82/04/24     82/05/06  17:04            99           193            10

   TOTALS:                                                           287           390            21

  END OF MEMBER LIST
```

Figure 113. Sample Index Listing - Source Library



```
TIME           *** SPF TRANSACTION LOG ***                    USERID: FLAG3     DATE: 82/05/28    PAGE: 1

17:11   START OF ISPF SESSION ----- SESSION #5 -----------------------------------------------------
17:12      UTILITY - SPECIFY  - LIBRARY 'SPFDEMO.NEWLIB.PLI' CREATED ON FLAG3 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 FLAG3 191.
17:19      BATCH   - JOBSTEP  - PLI OPT COMPILE -- TO SPFDEMO.NEWLIB.TEXT(COINS)
17:19      BATCH   - JOB      - FLAG3 U101  SUBMITTED.
17:20   END OF ISPF SESSION ------- SESSION #5 -----------------------------------------------------
```

Figure 114. Sample Log Listing

APPENDIX D. CHARACTER TRANSLATIONS FOR APL AND TEXT KEYBOARDS

PDF supports APL keyboards for all models of 3270 terminals, and text keyboards for 3278 and 3279 terminals. The 2-byte transmission codes for APL and text characters are translated by PDF into 1-byte codes for internal storage as shown in Figure 115 and Figure 116.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00																	0F
10																	1F
20																	2F
30																	3F
40	sp	A̲	B̲	C̲	D̲	E̲	F̲	G̲	H̲	I̲	¢	.	<	(	+	|	4F
50	&	J̲	K̲	L̲	M̲	N̲	O̲	P̲	Q̲	R̲	!	$	*	)	;	¬	5F
60	–	/	S̲	T̲	U̲	V̲	W̲	X̲	Y̲	Z̲	¦	,	%	_	>	?	6F
70		∧	¨						∨	`	:	#	@	'	=	"	7F
80	∼	a	b	c	d	e	f	g	h	i	↑	↓	≤	⌈	⌊	→	8F
90	⎕	j	k	l	m	n	o	p	q	r	⊃	⊂		○		←	9F
A0	¯	∼	s	t	u	v	w	x	y	z	∩	∪		[	≥	∘	AF
B0	⍺	∊	⍳	⍴	⍵		×	\	÷		∇	∆	⊤	]	≠	∣	BF
C0	{	A	B	C	D	E	F	G	H	I	⍲	⍱		⌽		⍉	CF
D0	}	J	K	L	M	N	O	P	Q	R	⌶	!	⍒	⍋	⍞	⍝	DF
E0	\		S	T	U	V	W	X	Y	Z	⊣	⊢		⊖	⌹	⍕	EF
F0	0	1	2	3	4	5	6	7	8	9		⍫	⍙	●	⍎		FF

◤ 3278 only; invalid character on 3277.

◢ National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Figure 115. Internal Character Representations for APL Keyboards

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00																	0F
10																	1F
20																	2F
30																	3F
40	sp										¢	.	<	(	+	|	4F
50	&	$_1$	$_2$	$_3$						↓	!	$	*	)	;	¬	5F
60	-	/									¦	,	%	_	>	?	6F
70	$_n$	°								\`	:	#	@	'	=	"	7F
80		a	b	c	d	e	f	g	h	i	↑	{	≤	(	+	+	8F
90	□	j	k	l	m	n	o	p	q	r		}	¤	)	±	■	9F
A0	¯	∿	s	t	u	v	w	x	y	z	●	└	┌	[	≥	•	AF
B0	^{0}	^{1}	^{2}	^{3}	^{4}	^{5}	^{6}	^{7}	^{8}	^{9}	∇	┘	┐	]	≠	—	BF
C0	{	A	B	C	D	E	F	G	H	I	Δ	⊥	⊤	∨	⍀	⌿	CF
D0	}	J	K	L	M	N	O	P	Q	R	⌹	⍙	§	¶	←	→	DF
E0	\	\	S	T	U	V	W	X	Y	Z	⍫	⊦	⊣	⍋	⍒	∧	EF
F0	0	1	2	3	4	5	6	7	8	9	|	⌊	Φ	Φ	⌈		FF

□ National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Figure 116. Internal Character Representations for Text Keyboards

APPENDIX E. SUMMARY OF COMMANDS

GENERAL COMMANDS

Following is a description of the general PDF commands. Additional commands are defined for member lists, browse, edit, and dialog test. The default PF key definitions are shown following the command name.

HELP (PF1/13) Displays additional information about an error message or provides tutorial information about PDF commands and options.

SPLIT (PF2/14) Causes split screen mode to be entered, or changes the location of the split line.

END (PF3/15) Terminates the current operation and returns to the previous panel. If the primary option menu is displayed, this command terminates PDF.

RETURN (PF4/16) Causes an immediate return to the primary option menu or to the panel from which a HELP or KEYS command was entered, without displaying intervening panels.

RFIND (PF5/17) Repeats the action of the previous FIND command or the FIND part of the most recent CHANGE command (applies to browse and edit only).

RCHANGE (PF6/18) Repeats the action of the previous CHANGE command (applies to edit only).

UP (PF7/19) Causes a scroll toward the top of the data.

DOWN (PF8/20) Causes a scroll toward the bottom of the data.

SWAP (PF9/21) Moves the cursor to wherever it was previously positioned on the other logical screen of a split screen pair.

LEFT (PF10/22) Causes a scroll toward the left margin of the data.

RIGHT (PF11/23) Causes a scroll toward the right margin of the data.

CURSOR (PF12/24) Moves the cursor to the first input field on line 2 (normally the option selection or command field). Entering this command again causes the cursor to be moved to the second input field on line 2, if any (normally the scroll amount field).

PRINT — Causes a "snapshot" of the screen image to be recorded in the ISPF list file, for subsequent printing.

PRINT-HI — Same as PRINT except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. Do not use this command if the list output is to be spooled to a printer that does not have a suppress-space carriage control.

CP or CMS — Allows you to enter a CP or CMS command or an EXEC from any panel.

KEYS — Causes an immediate display of a panel that allows you to view and change the current PF key definitions (equivalent to option 0.3).

PANELID — The command **PANELID** or **PANELID ON** causes all subsequent panels to be displayed with the identifier of the panel shown at the beginning of line 1 (provided the panel contains a protected-field attribute byte in row 1 column 1, and no other attribute bytes in the next eight character positions). The command **PANELID OFF** suppresses display of the panel identifiers. The OFF mode is in effect at the beginning of a PDF session.

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, foreground, or batch (invalid on 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.

B (browse) — Causes the member to be browsed (valid only under the library utility).

D (delete) Causes the member to be deleted (valid only 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.

S (select) Causes the member to be selected for browse, edit, foreground, batch, and for move/copy, reset, or convert utility (invalid 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.

CAPS Causes character strings, entered as part of the FIND command, to be converted to uppercase. CAPS OFF causes character strings to be left as-is.

COLS Causes display of a line that identifies column positions.

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.

HEX Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.

LOCATE Causes direct scrolling to the specified relative line number or label.

RESET Removes the column identification line from display.

.xxxxx Establishes a label (PDF 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.

EDIT PRIMARY COMMANDS

Under edit, the following commands may be entered in the command/option field (line 2 of the logical screen).

AUTOLIST Causes a source listing of the edited data to be automatically recorded in the ISPF list file whenever edit is terminated by entering the END command. AUTOLIST OFF bypasses automatic source listings.

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.

CANCEL Causes editing to be terminated without saving the data.

CAPS Causes alphabetic data entered from the terminal to be translated to uppercase, including FIND and CHANGE strings. CAPS OFF causes alphabetic data to be left as-is.

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 ISPF library or MACLIB, or a sequential CMS file.

CREATE Causes a line or block of lines to be stored in any ISPF 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.

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.

HEX Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.

LOCATE Causes direct scrolling to the specified line number.

MODEL Causes model data to be copied from a skeleton library and inserted at a point indicated with the A (after) or B (before) line command.

MOVE Same as COPY, except that the other source is deleted after the copy operation.

NULLS Replaces trailing blanks on the screen with null characters. NULLS OFF causes trailing blanks to be sent to the screen as blanks.

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.

PROFILE Causes the display of the current setting of edit modes (NUMBER, AUTONUM, STATS, AUTOLIST, CAPS, NULLS, TABS, HEX, and RECOVERY). Also displays the MASK, TABS, and BOUNDS lines whenever they have settings other than their defaults.

RECOVERY Enables the PDF edit recovery function. If there is a subsequent system failure, the user can recover the PDF edit session up to the point of failure. RECOVERY OFF disables the edit recovery function.

RENUM Renumbers each line and turns on number mode.

REPLACE Same as CREATE, except that it allows an existing member or a sequential CMS file to be replaced.

RESET Causes a general resetting of intensified messages, incomplete line commands, and special lines.

SAVE Causes the data to be stored back into the edit file and editing to continue.

STATS Causes statistics to be generated or updated whenever a member is saved, created, or replaced. STATS OFF causes members to be stored without statistics.

SUBMIT Causes the data being edited to be punched to another virtual machine for batch execution.

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.

UNNUM Causes sequence numbers to be set to blanks and turns off number mode.

EDIT LINE COMMANDS

Under edit, line commands may be entered at the beginning of a line by overtyping the line number.

A (after) Identifies the line after which copied, moved, or model lines are to be inserted.

B (before) Identifies the line before which copied, moved, or model lines are to be inserted.

BOUNDS Causes display of the column boundary line, to allow redefinition of boundary positions.

COLS Causes display of a line that identifies column positions.

C (copy) Identifies a line to be copied.

D (delete) Causes the line to be deleted.

F (first) Causes the first line in a block of excluded lines to be redisplayed.

I (insert) Causes a new line to be inserted after this line.

L (last) Causes the last line in a block of excluded lines to be redisplayed.

MASK Causes display of the insert mask, to allow pre-entry of data on inserted lines.

M (move) Identifies a line to be moved.

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 overlaid with corresponding characters in the source line(s).

R (repeat) Causes the line to be repeated and the duplicate copy to be inserted after this line.

S (show) Causes an excluded line 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.

TF Causes reflowing of a paragraph of text from the cursor position to the end of the paragraph.

TS Causes splitting of a line into two lines at the designated cursor position to facilitate the insertion of new material in existing text.

X (exclude) Causes the line to be excluded from display. May also be used to limit the scope of the FIND and CHANGE commands.

< (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.

The following line commands may be entered with double characters to indicate the beginning and end of a block of lines.

CC	- Block copy
DD	- Block delete
MM	- Block move
OO	- Block overlay
RR	- Block repeat
XX	- Block exclude
<<	- Block "data" shift left
>>	- Block "data" shift right
((	- Block "column" shift left
))	- Block "column" shift right

The following line commands may incorporate a number (n) consisting of one or more digits.

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
Cn	- Copy n lines starting at this line
Dn	- Delete n lines starting at this line
Fn	- Show first n lines in a block of excluded lines
In	- Insert n lines following this line
Ln	- Show last n lines in a block of excluded lines
Mn	- Move n lines starting at this line
On	- Overlay n lines
Rn	- Repeat this line n times
RRn	- Repeat block of lines n times
Sn	- Show n lines in a block of excluded lines
TEn	- Insert n text entry lines following this line
TFn	- Flow text from left bound to column position n
TSn	- Insert n new lines between the split lines
Xn	- Exclude n lines starting at 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

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 current right boundary)

DIALOG TEST PRIMARY COMMANDS

The following primary commands have special meaning during dialog test operations. They are entered in the command line of the applicable dialog test option panels.

CANCEL (CAN) Ends the current option. Any changes made to the data are ignored.

END Ends the current option. Any changes made to the data now take effect.

LOCATE (LOC, L) argument Positions the scrollable display to the first (next) row containing the argument. The scan starts at the end of the first row currently being displayed. A message is displayed indicating the result of the scan.

QUAL This command is entered only from the breakpoints panel. It displays to the user the breakpoint qualification data.

RESUME (RES) This command is entered on the qualification panel. It restores the format of the breakpoints panel. Each breakpoint that has qualification is flagged by the characters "*QUAL*" in columns 75 to 80 on that line of the breakpoints panel.

DIALOG TEST LINE COMMANDS

The following line commands have special meaning during testing operations:

D or Dn Delete one line or n lines starting with this line.

I or In Insert one line or n lines directly after this line (with underscores and quotes in the appropriate fields).

R or Rn Repeat this line once or n times.

APPENDIX F. SUMMARY OF COMMAND SYNTAX

MEMBER LIST COMMANDS

Primary Commands:

```
LOCATE  member-name

SELECT  member-name
```

Line Selection Commands:

```
S (select) - except option 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

```
CAPS  [ON ]
      [OFF]

COLS

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

HEX   [ON ] [VERT]
      [OFF] [DATA]

LOCATE  line-number
        label

.xxxxx

RESET
```

EDIT PRIMARY COMMANDS

```
AUTOLIST  [ON ]
          [OFF]

AUTONUM  [ON ]
         [OFF]

CANCEL
CAN

CAPS  [ON ]
      [OFF]

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

COPY  [member-name]

CREATE  [member-name]

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

HEX  [ON ] [VERT]
     [OFF] [DATA]

LOCATE  line-number

MODEL [[class-name.] model-name [.qualifier...]]
      [CLASS [class-name]                      ]

MOVE  [member-name]

NULLS  [ON ] [ALL]
       [OFF]

NUMBER  [ON ] [STD] [COBOL] [DISPLAY]
        [OFF]

PROFILE  [name] [number]

RECOVERY  [ON ]
          [OFF]

RENUM   [STD] [COBOL] [DISPLAY]
```

```
REPLACE  [member-name]

RESET

SAVE

STATS  [ON ]
       [OFF]

SUBMIT  machine-id

TABS   [ON ] [tab-char] [ALL]
       [OFF]

UNNUM
```

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 nonblank character
P'.'	-	any non-displayable (invalid) character
P'#'	-	any numeric character (0-9)
P'-'	-	any non-numeric character
P'@'	-	any alphabetic character (uppercase or lowercase)
P'<'	-	any lowercase alphabetic character
P'>'	-	any uppercase alphabetic character
P'$'	-	any special character (not alphabetic or numeric)

If an APL or TEXT keyboard is being used, the following additional character is defined:

P'⍠' - any APL-specific or TEXT-specific character

EDIT LINE COMMANDS

A, An	(after)
B, Bn	(before)
BOUNDS	
COLS	
C, Cn, CC	(copy)
D, Dn, DD	(delete)
F, Fn	(first)
I, In	(insert)
L, Ln	(last)
MASK	
M, Mn, MM	(move)
O, On, OO	(overlay)
R, Rn, RR, RRn	(repeat)
S, Sn	(show)
TABS	
TE, TEn	(text entry)
TF, TFn	(text flow)
TS, TSn	(text split)
X, Xn, XX	(exclude)
<, <n, <<, <<n	(data left)
>, >n, >>, >>n	(data right)
(, (n, ((, ((n	(cols left)
),)n,)),))n	(cols right)

APPENDIX G. DIALOG DEVELOPMENT MODEL LISTING

This appendix contains a listing of the models shipped with PDF. The table shows the external model name, any qualifiers, a short description, and the internal member name in the SKELS library shipped with PDF. Refer to "Dialog Development Models" in Chapter 6 for information on adding or changing a model.

ISPF SERVICES IN CLISTS

Table Functions (General):

Model Name	Qualifier	Description	Member Name
TBCLOSE		Table close	ISREMCG5
TBCREATE		Table create	ISREMCG1
TBEND		Table end	ISREMCG6
TBERASE		Table erase	ISREMCG7
TBOPEN		Table open	ISREMCG2
TBQUERY		Table query	ISREMCG3
TBSAVE		Table save	ISREMCG4

Table Functions (Row):

TBADD		Table row add	ISREMCR1
TBBOTTOM		Table row pointer to bottom	ISREMCRA
TBDELETE		Table delete	ISREMCR2
TBEXIST		Table exist	ISREMCR6
TBGET		Table get	ISREMCR3
TBMOD		Table modify	ISREMCR5
TBPUT		Table put	ISREMCR4
TBSARG		Table search argument	ISREMCR7
TBSCAN		Table scan	ISREMCR8
TBSKIP		Table skip	ISREMCRA
TBTOP		Table top	ISREMCR9
TBVCLEAR		Variable clear	ISREMCRC

Variables:

VGET		Variable get	ISREMCV1
VPUT		Variable put	ISREMCV2

Model Name	Qualifier	Description	Member Name
File Tailoring:			
FTCLOSE		End file tailoring	ISREMCF3
FTERASE		File tailor erase	ISREMCF4
FTINCL		File tailor include skeleton	ISREMCF2
FTOPEN		File tailor open	IREMMCF1
Display:			
DISPLAY		Display option	ISREMCD1
SETMSG		Set message display	ISREMCD3
TBDISPL		Table display information	ISREMCD2
Miscellaneous:			
BROWSE		Browse service	ISREMCM3
CONTROL		Control service	ISREMCM2
EDIT		Edit service	ISREMCM4
LOG		Write message or log file	ISREMCM5
SELECT		Select service	ISREMCM1

ISPF SERVICES IN COBOL PROGRAMS

Model Name	Qualifier	Description	Member Name
Working Storage Section:			
WORKSTOR		Working storage definition	ISREMBW1
Table Functions (General):			
TBCLOSE		Table close	ISREMBG5
TBCREATE		Table create	ISREMBG1
TBEND		Table end	ISREMBG6
TBERASE		Table erase	ISREMBG7
TBOPEN		Table open	ISREMBG2
TBQUERY		Table query	ISREMBG3
TBSAVE		Table save	ISREMBG4
Table Functions (Row):			
TBADD		Table row add	ISREMBR1
TBBOTTOM		Table row pointer to bottom	ISREMBRA
TBDELETE		Table delete	ISREMBR2
TBEXIST		Table exist	ISREMBR6
TBGET		Table get	ISREMBR3
TBPUT		Table put	ISREMBR4
TBMOD		Table modify	ISREMBR5
TBSARG		Table search argument	ISREMBR7
TBSCAN		Table scan	ISREMBR8

Model Name	Qualifier	Description	Member Name
TBSKIP		Table skip	ISREMBRB
TBTOP		Table top	ISREMBR9
TBVCLEAR		Variable clear	ISREMBRC
Variables:			
VCOPY		Copy variable	ISREMBV5
VDEFINE		Variable define	ISREMBV3
VDELETE		Variable delete	ISREMBV4
VGET		Variable get	ISREMBV1
VPUT		Variable put	ISREMBV2
VREPLACE		Variable replace	ISREMBV6
VRESET		Variable reset	ISREMBV7
File Tailoring:			
FTCLOSE		End file tailoring	ISREMBF3
FTERASE		File tailor erase	ISREMBF4
FTINCL		File tailor include skeleton	ISREMBF2
FTOPEN		File tailor open	IREMMBF1
Display:			
DISPLAY		Display option	ISREMBD1
SETMSG		Set message display	ISREMBD3
TBDISPL		Table display information	ISREMBD2
Miscellaneous:			
BROWSE		Browse service	ISREMBMB
BROWSE	MVS	Browse service (MVS)	ISREMCB1
BROWSE	VM	Browse service (VM)	ISREMCB2
CONTROL		Control service	ISREMBM2
EDIT		Edit service	ISREMBME
EDIT	MVS	Edit service (MVS)	ISREMCE1
EDIT	VM	Edit service (VM)	ISREMCE2
LOG		Write message or log file	ISREMBM5
SELECT		Select service	ISREMBM1

ISPF SERVICES IN EXECS

Table Functions (General):

TBCLOSE	Table close	ISREMEG5
TBCREATE	Table create	ISREMEG1
TBEND	Table end	ISREMEG6
TBERASE	Table erase	ISREMEG7
TBOPEN	Table open	ISREMEG2
TBQUERY	Table query	ISREMEG3
TBSAVE	Table save	ISREMEG4

Table Functions (Row):

TBADD	Table row add	ISREMER1
TBBOTTOM	Table row pointer to bottom	ISREMERA
TBDELETE	Table delete	ISREMER2
TBEXIST	Table exist	ISREMER6
TBGET	Table get	ISREMER3
TBMOD	Table modify	ISREMER5
TBPUT	Table put	ISREMER4
TBSARG	Table search argument	ISREMER7
TBSCAN	Table scan	ISREMER8
TBSKIP	Table skip	ISREMERA
TBTOP	Table top	ISREMER9
TBVCLEAR	Variable clear	ISREMERC

Variables:

VGET	Variable get	ISREMEV1
VPUT	Variable put	ISREMEV2

File Tailoring:

FTCLOSE	End file tailoring	ISREMEF3
FTERASE	File tailor erase	ISREMEF4
FTINCL	File tailor include skeleton	ISREMEF2
FTOPEN	File tailor open	IREMMCF1

Display:

DISPLAY	Display option	ISREMED1
SETMSG	Set message display	ISREMED3
TBDISPL	Table display information	ISREMED2

Model Name	Qualifier	Description	Member Name
Miscellaneous:			
BROWSE		Browse service	ISREMEM3
CONTROL		Control service	ISREMEM2
EDIT		Edit service	ISREMEM4
LOG		Write message or log file	ISREMEM5
SELECT		Select service	ISREMEM1

ISPF SERVICES IN FORTRAN PROGRAMS

Table Functions (General):

TBCLOSE	Table close	ISREMFG5
TBCREATE	Table create	ISREMFG1
TBEND	Table end	ISREMFG6
TBERASE	Table erase	ISREMFG7
TBOPEN	Table open	ISREMFG2
TBQUERY	Table query	ISREMFG3
TBSAVE	Table save	ISREMFG4

Table Functions (Row):

TBADD	Table row add	ISREMFR1
TBBOTTOM	Table row pointer to bottom	ISREMFRA
TBDELETE	Table delete	ISREMFR2
TBEXIST	Table exist	ISREMFR6
TBGET	Table get	ISREMFR3
TBMOD	Table modify	ISREMFR5
TBPUT	Table put	ISREMFR4
TBSARG	Table search argument	ISREMFR7
TBSCAN	Table scan	ISREMFR8
TBSKIP	Table skip	ISREMFRB
TBTOP	Table top	ISREMFR9
TBVCLEAR	Variable clear	ISREMFRC

Variables:

VCOPY	Copy variable	ISREMFV5
VDEFINE	Variable define	ISREMFV3
VDELETE	Variable delete	ISREMFV4
VGET	Variable get	ISREMFV1
VPUT	Variable put	ISREMFV2
VREPLACE	Variable replace	ISREMFV6
VRESET	Variable reset	ISREMFV7

Model Name	Qualifier	Description	Member Name
File Tailoring:			
FTCLOSE		End file tailoring	ISREMFF3
FTERASE		File tailor erase	ISREMFF4
FTINCL		File tailor include skeleton	ISREMFF2
FTOPEN		File tailor open	IREMMFF1
Display:			
DISPLAY		Display option	ISREMFD1
SETMSG		Set message display	ISREMFD3
TBDISPL		Table display information	ISREMFD2
Miscellaneous:			
BROWSE		Browse service	ISREMFMB
BROWSE	MVS	Browse service (MVS)	ISREMFB1
BROWSE	VM	Browse service (VM)	ISREMFB2
CONTROL		Control service	ISREMFM2
EDIT		Edit service	ISREMFME
EDIT	MVS	Edit service (MVS)	ISREMFE1
EDIT	VM	Edit service (VM)	ISREMFE2
LOG		Write message or log file	ISREMFM5
SELECT		Select service	ISREMFM1
MESSAGE FORMAT			
MSGS		Message member selection	ISREMMSG
PANEL FORMATS AND STATEMENTS			
Panel Formats:			
ENTRY		Data entry panel	ISREMMF1
MULTIPLE		Multiple-column panel	ISREMSEL
MULTIPLE	SELECT2	Double-column selection panel	ISREMSE1
MULTIPLE	ENTRY2	Double-column entry panel	ISREMSE2
SELECTION		Menu/selection panel	ISREMMF3
TBDISPL		Table display table	ISREMMF4
TUTORIAL		Help/tutorial panel	ISREMMF5

Model Name	Qualifier	Description	Member Name
Panel Statements:			
ASSIGN		Assignment statement	ISREMASG
ASSIGN	SIMPLE	Simple assignment statement	ISREMAS1
ASSIGN	TRANS	Trans assignment statement	ISREMAS2
ASSIGN	TRUNC	Trunc assignment statement	ISREMAS3
ASSIGN	TRANSTRU	Nested translate truncate statement	ISREMAS4
ATTR		Attribute section header	ISREMMS2
ATTRIB		New attribute character definition	ISREMMS3
BODY		Body section header	ISREMMS4
CONTROL		Control variables	ISREMCNL
CONTROL	CURSOR	Control initial cursor placement	ISREMCN1
CONTROL	HELP	Establish a tutorial panel	ISREMCN2
CONTROL	MSG	Identify message to be displayed	ISREMCN3
CONTROL	RESP	Indicate user response to panel	ISREMCN4
CONTROL	TRAIL	Contain remainder from TRUNC function	ISREMCN5
IF		IF statement	ISREMMS6
MODEL		Model section header	ISREMMS7
VERIFY		Verification statements	ISREMVER
VERIFY	ALPHA	Alphabetic characters	ISREMVE1
VERIFY	BIT	Binary characters	ISREMVE2
VERIFY	DSNAME	TSO data set name	ISREMVE3
VERIFY	FILEID	CMS fileid	ISREMVE4
VERIFY	HEX	Hexadecimal characters	ISREMVE5
VERIFY	LIST	List of valid values	ISREMVE6
VERIFY	NAME	Data set member name	ISREMVE7
VERIFY	NONBLANK	Verify nonblank field	ISREMVE8
VERIFY	NUM	Numeric characters	ISREMVE9
VERIFY	PICT	Mixed characters matching picture	ISREMVEA
VERIFY	RANGE	Numeric value within specified limits	ISREMVEB
VPUT		Variable put statement	ISREMMS9

ISPF SERVICES IN PL/I PROGRAMS

Table Functions (General):

TBCLOSE		Table close	ISREMPG5
TBCREATE		Table create	ISREMPG1
TBEND		Table end	ISREMPG6
TBERASE		Table erase	ISREMPG7
TBOPEN		Table open	ISREMPG2
TBQUERY		Table query	ISREMPG3
TBSAVE		Table save	ISREMPG4

Model Name	Qualifier	Description	Member Name
File Tailoring:			
FTCLOSE		End file tailoring	ISREMPF3
FTERASE		File tailor erase	ISREMPF4
FTINCL		File tailor include skeleton	ISREMPF2
FTOPEN		File tailor open	IREMMPF1
Table Functions (Row):			
TBADD		Table row add	ISREMPR1
TBBOTTOM		Table row pointer to bottom	ISREMPRA
TBDELETE		Table delete	ISREMPR2
TBEXIST		Table exist	ISREMPR6
TBGET		Table get	ISREMPR3
TBMOD		Table modify	ISREMPR5
TBPUT		Table put	ISREMPR4
TBSARG		Table search argument	ISREMPR7
TBSCAN		Table scan	ISREMPR8
TBSKIP		Table skip	ISREMPRB
TBTOP		Table top	ISREMPR9
TBVCLEAR		Variable clear	ISREMPRC
Display:			
DISPLAY		Display option	ISREMPD1
SETMSG		Set message display	ISREMPD3
TBDISPL		Table display information	ISREMPD2
Variables:			
VCOPY		Copy variable	ISREMPV5
VDEFINE		Variable define	ISREMPV3
VDELETE		Variable delete	ISREMPV4
VGET		Variable get	ISREMPV1
VPUT		Variable put	ISREMPV2
VREPLACE		Variable replace	ISREMPV6
VRESET		Variable reset	ISREMPV7
Miscellaneous:			
BROWSE		Browse service	ISREMPMB
BROWSE	MVS	Browse service (MVS)	ISREMPB1
BROWSE	VM	Browse service (VM)	ISREMPB2
CONTROL		Control service	ISREMPM2
EDIT		Edit service	ISREMPME
EDIT	MVS	Edit service (MVS)	ISREMPE1
EDIT	VM	Edit service (VM)	ISREMPE2
LOG		Write message or log file	ISREMPM5
SELECT		SELECT service	ISREMPM1

Model Name	Qualifier	Description	Member Name
FILE TAILORING CONTROL STATEMENTS			
CM		Define comment statement	ISREMSK1
DEFAULT		Change control character defaults	ISREMSK2
DOT		Define DO group for table row	ISREMSK3
IM		Imbed specified file skeleton	ISREMSK4
SEL		Conditional execution definition	ISREMSK5
SET		Set dialog variable value	ISREMSK6
TB		Set tab stop position	ISREMSK7

INDEX

Interactive
System Productivity Facility/
Program Development Facility
for VM/SP: Program Reference

SC34-2090-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It 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 comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

____

____

____

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? ______________________________

Number of latest Technical Newsletter (if any) concerning 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.)

SC34-2090-0

Reader's Comment Form

Cut or Fold Along Line

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 Z59, Building 931
P. O. Box 390
Poughkeepsie, New York 12602

Fold and tape

Please Do Not Staple

Fold and tape

IBM®

Interactive
System Productivity Facility/
Program Development Facility
for VM/SP: Program Reference SC34-2090-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It 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 comments 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 Technical Newsletter (if any) concerning 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.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

SC34-2090-0

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 Z59, Building 931
P. O. Box 390
Poughkeepsie, New York   12602

Fold and tape   **Please Do Not Staple**   Fold and tape

Cut or Fold Along Line

ISPF/PDF for VM/SP: Program Reference   Printed in U.S.A.   SC34-2090-0

SC34-2090-0

ISPF/PDF for VM/SP: Program Reference   Printed in U.S.A.   SC34-2090-0

SC34-2090-0