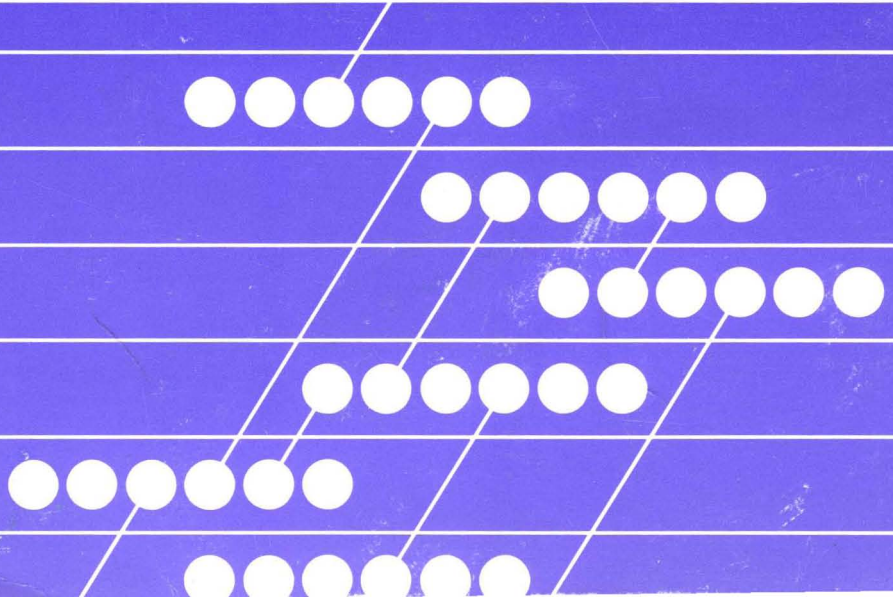


Interactive
System Productivity Facility/
Program Development Facility
Version 2 Release 2

Program Reference

MVS

IBM



Interactive
System Productivity Facility/
Program Development Facility
Version 2 Release 2

Program Reference

→ Prior version was SC34-~~462~~²⁰⁸⁹

MVS

Publication Number
SC34-4024-1

File Number
S370/4300-39

Program Number
5665-317

Second Edition (September 1985)

This edition applies to Version 2, Release 2 of the program product Interactive System Productivity Facility/Program Development Facility (5665-317), and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. It is for use with the Interactive System Productivity Facility Version 2, Release 2 (5665-319), and OS/VS2 (MVS) Release 3.8 or later. In the Extended Architecture (XA) environment, it is for use with MVS/SP Version 2.

Changes are made periodically to the information herein. Therefore, before using this publication, consult the latest IBM System/370 and 4300 Processors Bibliography (GC20-0001) for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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.

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, Dept. T46, P. O. Box 60000, Cary, NC 27511. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

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 the 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 of a programmer's tasks.

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

The first three chapters of this manual provide you with information pertinent to the general use of PDF, such as use of the terminal, format of the displays, and a description of the programming libraries. 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 and ISPF/PDF for MVS General Information, GC34-4041. You may also want to refer to the following publications:

ISPF for MVS Dialog Management Services (SC34-4021) - Provides a detailed description of the dialog management services and related information required to develop an interactive application that runs under ISPF.

ISPF for MVS Dialog Management Services Examples (SC34-4022) - Provides a set of examples of the use of dialog management services.

ISPF and ISPF/PDF for MVS Installation and Customization (SC34-4019) - Provides information needed to install ISPF and ISPF/PDF and to custom tailor these products for a particular installation.

ISPF/PDF for MVS Library Management (SC34-4025) - Provides information needed to use the library management facilities provided with PDF.

ISPF/PDF for MVS Edit Macros (SC34-4018) - Provides information needed to write and test macros to be executed under the PDF editor.

ISPF/PDF for MVS Services (SC34-4023) - Provides a detailed description of the EDIT, BROWSE, and library access dialog services provided with PDF.

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 the OR symbol (|) or 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 separated by the OR symbol (|) or are stacked and shown in braces { }, you must select one of the choices.
- Command name truncations are shown stacked under the full command name without braces.

CONTENTS

Chapter 1. Introduction	1
Major Functions	1
Structured Programming	2
Interactive Applications	2
Usage Scenario	3
Chapter 2. Interacting with PDF	5
Panel Displays	5
Format of Panels	5
Types of Panels	6
PDF Invocation	10
Command Entry	13
Jump Function	16
Terminal Keys	17
Program Access Keys	17
Program Function Keys	18
Scrolling	19
Overtyping of Scroll and Multiple Choice Parameters	20
Split Screen	21
Help Information	22
PDF Termination	23
Chapter 3. Libraries and Data Sets	25
ISPF Libraries	25
Library Statistics	28
Load Module Libraries	30
Parameter Specification	31
Library and Data Set Names	31
Data Set Password	32
Job Statement Information	33
User Profiles	33
Member List Commands	34
Locating Members	35
Single-Selection Member Lists	35
Multiple-Selection Member Lists	36
Log and List Files	36
Foreground and Batch Output Listings	37
Other Temporary Files	38
Chapter 4. ISPF Parms (Option 0)	39
Specify Terminal Characteristics (Option 0.1)	40
Specify Log and List Defaults (Option 0.2)	43
Specify Program Function Keys (Option 0.3)	45
Specify Screen Display Characteristics (Option 0.4)	49
Specify List Data Set Characteristics (Option 0.5)	50
Chapter 5. Browse (Option 1)	53

Locating Lines	57
Identifying Columns	58
Finding Character Strings	59
Controlling the Display (DISPLAY Command)	61
Displaying Data in Hexadecimal Format	62
Recursive Browsing (BROWSE Command)	63
Chapter 6. Edit (Option 2)	65
Disabling Library Management	67
Edit Selection	68
Summary of Edit Operations	72
Command Functions	73
Edit Modes, Options, and Profiles	74
Selecting a Profile	75
Setting Up a Profile	75
Locking Your Profile	76
Special Data Modes	76
Packing Data	77
Macros	77
Labels and Line Ranges	78
Line Ranges	79
Edit Termination	81
Primary Commands and Functions	82
Controlling Your Edit Environment	82
Sequence Number Generation and Control	83
Sequence Number Format	83
Sequence Number Display	84
Initialization of Number Mode	84
NUMBER Command	85
RENUM Command	85
UNNUM Command	86
AUTONUM Command	86
Statistics Generation and Control (STATS, VERSION, and LEVEL Commands)	86
Controlling the END Command (AUTOSAVE Command)	88
Packing Data (PACK Command)	89
Automatic Source Listing (AUTOLIST Command)	89
Character Translation (CAPS Command)	90
Nulls Control (NULLS Command)	90
Tabs Definition (TABS Primary Command)	91
Profile Display and Control (PROFILE Command)	93
Hexadecimal Display (HEX Command)	95
Controlling the Edit Boundaries (BOUNDS Command)	96
Displaying Model Notes (NOTE Command)	98
General Reset (RESET Command)	99
Recovery Control (RECOVERY Command)	100
Macro-Handling Commands	101
Specifying an Initial Macro (IMACRO Command)	101
Specifying a Recovery Macro (RMACRO Command)	102
Define a Name (DEFINE Command)	103
Execute a Built-in Command (BUILTIN Command)	104
Locating Lines (LOCATE Command)	104
Finding, Changing, and Excluding Data	106
Command Summary	106

Detailed Command Description	110
Starting Point, Direction, and Extent of Search	112
Conditions for Character String Match	113
Use of Text Strings	114
Use of Character Strings	114
Use of Picture Strings for String-1	114
Use of Picture Strings for String-2	116
Excluded Line Limitations	116
Column Limitations	116
Range of Lines Limitation	117
Example of Limiting Operands	117
Merging and Segmenting Data	117
Merging Data (COPY and MOVE Commands)	117
Segmenting Data (CREATE and REPLACE Commands)	120
Deleting Lines (DELETE Command)	122
Sorting Data (SORT Command)	123
Recursive Editing (EDIT Command)	125
Submitting to a Job Stream (SUBMIT Command)	125
Saving and Cancelling	126
Saving Data (SAVE Command)	126
Cancelling Changes (CANCEL Command)	127
Line Commands and Functions	127
Rules for Entering Line Commands	128
Basic Line Commands	129
I (Insert) Command	129
D (Delete) Command	131
R (Repeat) Command	132
M (Move), C (Copy), A (After), and B (Before) Commands	134
Preparing Text	136
TE (Text Entry) Command	137
TS (Text Split) Command	138
TF (Text Flow) Command	143
Uppercase (UC) Command	146
Lowercase (LC) Command	148
Overlaying Lines (O Command)	150
Identifying Columns (COLS Command)	152
Defining Masks (MASK Command)	153
Controlling Tabs (TABS Line Command)	155
Defining Boundary Columns (BOUNDS Command)	157
Changing Indentations	159
> and < (Data Shifting) Commands	161
) and ((Column Shifting) Commands	163
Excluding Lines (X, F, L, and S Commands)	164
Dialog Development Models	168
MODEL Command	171
Using Models	172
Adding, Changing, and Deleting Models	174
Adding Models	175
Finding Models	180
Changing Models	180
Deleting Models	180
Chapter 7. Utilities (Option 3)	181
Library Utility (Option 3.1)	182

Data Set Utility (Option 3.2)	186
Move/Copy Utility (Option 3.3)	192
Data Set List Utility (Option 3.4)	201
Reset Statistics Utility (Option 3.5)	206
Hardcopy Utility (Option 3.6)	209
List VTOC Utility (Option 3.7)	211
Outlist Utility (Option 3.8)	213
Command Table Utility (Option 3.9)	216
Convert Menus/Messages (Option 3.10)	219
Format Definition Utility (Option 3.11)	222
Chapter 8. Foreground (Option 4)	229
Member Expansion	229
Member Expansion Return Codes	231
COBOL Restrictions	231
Trigger Statement Errors	232
Processing Sequence	233
Assembler/Compiler Processing	234
Linkage Editor Processing	237
Script/VS Processing	239
Member Parts List	245
Chapter 9. Batch (Option 5)	249
Member Parts List	252
LIST Data Set	253
Chapter 10. Command (Option 6)	255
Chapter 11. Dialog Test (Option 7)	259
Dialog Test Environment	260
Dialog Test Variable Usage	261
Dialog Test Severe Error Handling	262
Dialog Test Commands	263
Primary Commands	264
Line Commands	265
Functions (Option 7.1)	266
Panels (Option 7.2)	269
Variables (Option 7.3)	272
Manipulating Variables	273
Variables Usage Notes	274
Tables (Option 7.4)	276
Display Row (Tables Option 1)	278
Delete Table Row (Tables Option 2)	280
Modify Row (Tables Option 3)	281
Add Row (Tables Option 4)	283
Display Structure (Tables Option 5)	285
Display Status (Tables Option 6)	287
Log (Option 7.5)	291
Trace Header	291
Function Trace Entries	292
Variable Trace Entries	292
Dialog Services (Option 7.6)	294
Traces (Option 7.7)	296
Function Trace (Trace Option 1)	297

Variable Trace (Trace Option 2)	298
Breakpoints (Option 7.8)	301
Defining Breakpoints (Breakpoints Panel)	301
Qualifying a Breakpoint (Qualification Panel)	303
Tutorial (Option 7.T)	306
Exit (Option 7.X)	307
Encountering a Breakpoint	308
Chapter 12. Library Management Utilities (Option 8)	311
Chapter 13. Tutorial (Option T)	313
Appendix A. Usage Scenario	317
Appendix B. PDF Listing Formats	323
Appendix C. Character Translations for APL and Text Keyboards	331
Appendix D. Summary of Commands	333
General Commands	333
Member List Commands	335
Browse Commands	335
Edit Primary Commands	336
Edit Line Commands	339
Dialog Test Primary Commands	341
Dialog Test Line Commands	342
Appendix E. Summary of Command Syntax	343
Member List Commands	343
Browse Commands	343
Edit Primary Commands	344
Find/Change Strings (Browse and Edit)	347
Edit Line Commands	347
Appendix F. Dialog Development Model Listing	349
Appendix G. List of Abbreviations	363
Index	365

FIGURES

1.	Utility Selection Panel	7
2.	Browse - Entry Panel	8
3.	Browse - Member List	9
4.	Browse - Data Display	10
5.	Primary Option Menu	11
6.	Default Program Key Arrangement	18
7.	Split Screen Example	22
8.	Termination Panel	24
9.	ISPF Library Organization	26
10.	Hierarchy of ISPF Libraries	27
11.	Member List Display	29
12.	Load Library Member List Display	30
13.	Parameter Selection Panel	39
14.	Terminal Characteristics Panel	40
15.	Log and List Defaults Panel	43
16.	PF Key Definition Panel (12 PF Keys)	45
17.	PF Key Definition Panels (24 PF Keys)	47
18.	Screen Display Characteristics	49
19.	List Data Set Characteristics Panel	50
20.	Browse - Entry Panel	53
21.	Browse - Data Display	56
22.	Browse - Column Identification Line	59
23.	Browse - Hexadecimal Display, Vertical	63
24.	Edit - Creating New Data	67
25.	Edit - LMF Error	68
26.	Edit - Entry Panel	69
27.	Edit - Member List	71
28.	Edit - Data Display	72
29.	Edit - Profile Display	94
30.	Edit - Hexadecimal Display, Data Representation	95
31.	Edit - Default Column Boundary Settings	98
32.	Edit - Recovery Panel	101
33.	Edit - CHANGE ALL Command Example	109
34.	Edit - COPY Command Example	119
35.	Edit - REPLACE Command Example	121
36.	Edit - Insert and Delete Lines Example	131
37.	Edit - Repeat Line Example	133
38.	Edit - Copy Lines Example	135
39.	Edit - Text Split Example	139
40.	Edit - Move Sentence Example	142
41.	Edit - Text Restructure Example	145
42.	Example of UC commands before processing	147
43.	Results of processing the UC commands shown in the previous figure	148
44.	Example of LC commands before processing	149
45.	Results of processing the LC commands shown in the previous figure	149
46.	Edit - Overlay Line Example	151

47.	Edit - Column Identification Line	153
48.	Edit - Define Mask Example	154
49.	Edit - Tab Definition Line	157
50.	Edit - Default Column Boundary Settings	158
51.	Edit - Boundary Definition Line	159
52.	Edit - Left Shift Data Example	162
53.	Edit - Excluded Lines Example	165
54.	Edit - Show Lines Example	167
55.	Edit - Model Class Selection Panel	170
56.	Edit - Model Selection Panel for CLIST Models	173
57.	Edit - Sample Dialog Development Model	174
58.	Edit - Sample Block Letter Model	175
59.	Edit - Panel Model Selection Panel	176
60.	Edit - Modified Visible Panel Model Selection Panel	177
61.	Edit - Modified)PROC Section of Panel Model Selection Panel	178
62.	Edit - Block Letter Model Selection Panel Coding	179
63.	Utility Selection Panel	181
64.	Library Utility Panel	182
65.	Library Utility - Print, Rename and Delete	184
66.	Data Set Utility Panel	186
67.	Data Set Utility - Allocate New Data Set	187
68.	Data Set Utility - Rename Data Set	189
69.	Data Set Utility - Display Information	191
70.	Move/Copy Utility Panels	192
71.	Promotion "To" Panel	195
72.	Move/Copy Utility - Copy Members from List	197
73.	Data Set List Utility Panel	201
74.	Data Set List Display (Short Format)	202
75.	Reset Statistics Utility Panel	206
76.	Hardcopy Utility Panel	209
77.	List VTOC Utility Panel	211
78.	Display VTOC Example	212
79.	Outlist Utility Panel	213
80.	Outlist Print Panel	215
81.	Command Table Utility Panel	216
82.	Command Table Editing Panel	217
83.	"From" Panel for Converting Menu/Message Definitions	220
84.	"To" Panel for Converting Menu/Message Definitions	221
85.	Format Specification Panel	222
86.	Format Definition Panel	224
87.	Copy Format Selection List Panel	225
88.	Format Selection List Panel	226
89.	Foreground Selection Panel	232
90.	Foreground - Assembler Example	233
91.	Foreground Print Panel	235
92.	Foreground - Linkage Editor Example	238
93.	SCRIPT/VS Utility - Selection Panel	239
94.	SCRIPT/VS Formatting Style Selection Panel	240
95.	SCRIPT/VS Options Specification Panel	241
96.	Example of specifying specific pages to print from a document	242
97.	Foreground Member Parts List Panel	245
98.	Member Parts List Display	247
99.	Batch Selection Panel	249
100.	Batch - PL/I Optimizing Compiler Example	250

101.	Batch Member Parts List Panel	253
102.	TSO Command Entry Panel	255
103.	Dialog Test - Primary Option Menu	260
104.	Dialog Test - Invoke Function Entry Panel	267
105.	Dialog Test - Display Panel Entry Panel	269
106.	Dialog Test - Message Display Panel	271
107.	Dialog Test - Variables Entry Panel	272
108.	Dialog Test - Tables Selection Panel	276
109.	Dialog Test - Display Table Row Entry Panel	279
110.	Dialog Test - Delete Table Row Confirmation Panel	280
111.	Dialog Test - Modify Table Row Entry Panel	281
112.	Dialog Test - Add Row Entry Panel	284
113.	Dialog Test - Table Structure Display	286
114.	Dialog Test - Table Status Display for a Closed Table	288
115.	Dialog Test - Table Status Display for an Open Table	290
116.	Dialog Test - Invoke Dialog Service Entry Panel	294
117.	Dialog Test - Special Display Panel	295
118.	Dialog Test - Traces Selection Panel	296
119.	Dialog Test - Function Trace Entry Panel	297
120.	Dialog Test - Variable Trace Entry Panel	299
121.	Dialog Test - Breakpoints Entry Panel	302
122.	Dialog Test - Qualification Panel	303
123.	Dialog Test - Breakpoint Primary Option Menu	308
124.	Library Management Utility Panel	312
125.	Tutorial - First Two Pages	313
126.	Sample Source Listing	325
127.	Sample Index Listing - Source Library	326
128.	Sample Index Listing - Load Library	327
129.	Sample Log Listing	328
130.	Sample Data Set List Listing	329
131.	Internal Character Representations for APL Keyboards	331
132.	Internal Character Representations for Text Keyboards	332

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 (four areas for 3290).
- **Utilities** to specify and maintain libraries and data sets.
- **Interface to standard language processors** (compilers, assemblers, and a linkage editor). These processors may be invoked either in the foreground or as 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. Most responses can be in your national language: for example, English, Kanji, or German.

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 data set name, member name, version number, and modification level. The short message area (line 1) is used to indicate:

- Current line (for browse) and column positions (for 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 overwrite 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 selection panels and data entry panels, contains column headings on member lists, and is treated as part of the data area on data displays. The long message overlays any other data on the line.

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 its number on the command/option line 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 the user has 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 data set
                Print, rename, delete, or browse members
                Compress data set
  2 DATASET  - Data set utility:
                Display data set information
                Allocate, rename, or delete entire data set
                Catalog or uncatalog data set
  3 MOVE/COPY - Move, copy, or promote members or data sets
  4 DSLIST   - Data set list:
                Print or display (to process) list of data set names
                Print or display VTOC information
  5 RESET    - Reset statistics for members of ISPF library
  6 HARDCOPY - Initiate hardcopy output
  7 VTOC     - Display or print VTOC entries for a DASD volume
  8 OUTLIST  - Display, delete, or print held job output
  9 COMMANDS - Create/change an application command table
 10 CONVERT  - Convert old format messages/menus to new format
 11 FORMAT   - Format definition for formatted data edit/browse

```

Figure 1. Utility Selection Panel

----- BROWSE - ENTRY PANEL -----

COMMAND ==>

ISPF LIBRARY:

PROJECT ==> SPFDEMO

GROUP ==> MYLIB ==> ==> ==>

TYPE ==> PLI

MEMBER ==> _ (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:

DATA SET NAME ==>

VOLUME SERIAL ==> (If not cataloged)

DATA SET PASSWORD ==> (If password protected)

MIXED MODE ==> (Specify YES or NO)

FORMAT NAME ==>

Figure 2. Browse - Entry Panel

```

BROWSE - SPFDEMO.MYLIB.PLI -----
COMMAND ==> -
  NAME          VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT        01.00   83/01/09   83/01/09 17:07    21    21    0  HUNTER
ACCT1          01.01   83/02/11   83/04/23 14:52   199   193    0  BECKETT
ACCT2          01.00   83/03/09   83/03/09 17:07    20    20    0  BECKETT
S COINS        01.04   83/04/24   83/04/28 16:20    19    19    4  JOSLIN
COMPX          01.00   83/01/21   83/01/21 11:08    44    44    0  JOSLIN
COMPY          01.01   83/01/14   83/01/16 12:30    13    13    1  FISHER
DCLS          01.00   83/04/23   83/04/23 15:14    20    20    0  LEESB
LISTNEW       01.02   83/04/23   83/05/06 10:00    17    13    6  LEESB
MAIN          01.00   83/01/09   83/01/09 17:08     4     4    0  KIRK
TESTDIR       01.02   83/04/23   83/05/06 09:04    30    43   10  LAIDLAW
UPDATE        01.00   83/03/26   83/04/01 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 *****
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 *****

```

Figure 4. Browse - Data Display

PDF INVOCATION

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 its number or letter in the option field and pressing the ENTER key; for example:

```
OPTION ==> 2
```

to select the edit option.

```

----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> _

0 ISPF PARMS - Specify terminal and user parameters      USERID - FLAG3
1 BROWSE     - Display source data or output listings   TIME    - 12:47
2 EDIT       - Create or change source data            TERMINAL - 3278
3 UTILITIES  - Perform utility functions               PF KEYS - 12
4 FOREGROUND - Invoke language processors in foreground
5 BATCH      - Submit job for language processing
6 COMMAND    - Enter TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES - Perform library management utility functions
C CHANGES   - Display summary of changes for this release
T TUTORIAL   - Display information about ISPF/PDF
X EXIT       - Terminate ISPF using list/log defaults

```

Enter END command to terminate ISPF.

Figure 5. Primary Option Menu

If you select options 0, 3, 4, 5, 7, or 8 you get another selection panel with a secondary list of options. As you get familiar with PDF, you may want to bypass the second selection panel by typing your first two selections (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 including your selection as a parameter of the PDF command. Examples:

```

PDF 2      - to go directly to edit
PDF 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 allocation parameters and default dispositions for log and list files.

- 1 BROWSE** To display source data or output listings. Browse is intended primarily for viewing large data sets 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 data set) into virtual storage and retains it there during edit operations.
- 3 UTILITIES** To print, rename, or delete library members or entire data sets; allocate data sets; move, copy, or promote data; display or print data set lists or VTOCs; reset library statistics; initiate hardcopy output; examine held SYSOUT data; create or modify application command tables; convert menus and messages to ISPF format from old SPF format; or define formats to be used with edit or browse.
- 4 FOREGROUND** To execute language processing programs in the foreground, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL/VS, linkage editor, COBOL or FORTRAN interactive debug, and the SCRIPT/VS document formatter.
- 5 BATCH** To generate and submit JCL for batch execution of IBM language processing programs, including: assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL, and linkage editor.
- 6 COMMAND** To enter a TSO command or command procedure (CLIST) 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.
- Privileged! (?)*
8 LM UTILITIES To control updating of ISPF libraries and to establish or remove updating controls.
- C CHANGES** To obtain online instruction about the changes between Version 1 and Version 2 of 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 log and list defaults defined usingParms 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 typing the command in the command/option 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 that has been equated to the desired command.

Although there is a set of default PF key definitions shipped with PDF, there are no functions that rely on specific PF key settings. 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 differentiate 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. However, if you type something unrelated to the PF key in the command field and then press a PF key, the entry in the command field will be passed through to the function in control. It will be up to the function in control to then process or ignore this entry. For example, Edit will process the entry and Tutorial will ignore the entry. Some PDF functions such as Edit and Browse will flag the entry in the command field as an invalid command if it is not valid for the function in control. Any stacked commands in the command field will be executed.

You can enter multiple commands in a single command entry. This is called command stacking. You 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). In the following example, two edit CHANGE commands have been stacked:

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

The PDF commands include the following functions:

HELP	SPLIT	END
RETURN	RFIND	RCHANGE
UP	DOWN	SWAP
LEFT	RIGHT	CURSOR
PRINT	PRINT-HI	TSO
KEYS	PANELID	SPLITV
PRINTL	PRINTLHI	

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 (applies to browse only).
- Finds the search string defined by the most recent SEEK, FIND, or CHANGE command, or excludes a line containing the search string defined by the previous EXCLUDE command (applies to 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. For use of the SWAP command, see the section entitled "Split Screen."

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). This command is useful only when invoked using a PF key.

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

Note: For the PRINT, PRINT-HI, PRINTL, and PRINTLHI commands, a screen image may exceed 121 characters. When it does, the line must be split when the output is being directed to a printer other than a 3800. The line length is obtained from a user modifiable specification on the ISPF PARMS panel (option 0.2). The default line length for printing is 121 characters.

PRINT-HI Same as PRINT except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. See the note under the PRINT command. Do not use this command if the list output is to be printed on a 328x printer or other device that does not have a suppress-space carriage control.

PRINTL Causes a "snapshot" of the logical screen image to be recorded in the ISPF list file for subsequent printing. See the note under the PRINT command.

PRINTLHI Same as PRINT except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. See the note under the PRINT command. Do not use this command if the list output is to be printed on a 328x printer or other device that does not have a suppress-space carriage control.

SPLITV On 3290 terminals, causes the screen to be split vertically.

Note: The SPLITV function is not active if the data being displayed on a screen is more than 80 characters wide.

TSO Allows you to enter a TSO command or CLIST from any panel. For example:

```
COMMAND ==> TSO LISTC LEVEL(Z77PHJ)
```

Note: If an ISPF dialog invoked using the TSO command encounters a severe error, you will be returned to the primary option menu, not the panel on which you entered the TSO command.

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 ten character positions). The command **PANELID OFF** suppresses display of the panel identifiers. **PANELID** is **OFF** 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 option under the same primary option menu without displaying the primary option menu, as follows: In the command field on any panel, or in any input field that can be displayed and is 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.

Notes:

1. 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 modeCommands stacked after =X in split screen mode will be ignored.
2. Since option 7 (dialog test) has its own primary option menu, 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.
3. 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 TSO commands, using the parms option (option 0.3) or the KEYS command.

Program Access Keys

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

ATTENTION (PA1) Normally, you should not use this key while you are in ISPF full screen mode.

If you press PA1 after the keyboard is unlocked by ISPF, it is treated like the PA2 key. It does not cause an exit from ISPF. However, if you press the PA1 key a second time without any intervening interaction, it causes current processing to be terminated and the primary option menu to be redisplayed.

If you press PA1 after manually unlocking the keyboard (with the RESET key), it causes current processing to be terminated and the primary option menu to be redisplayed.

RESHOW (PA2) Reshows the last full screen image displayed by ISPF. PA2 may be useful if you accidentally press the ERASE INPUT or CLEAR key, or if you type 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. Use option 0.3 to reset the program function keys for your use.

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 or during table display, 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.

If you are browsing or editing a formatted data set, scrolling left or right is done on a field basis, not a column basis. If you specify a scroll amount, the scroll amount will be ignored. Instead, scrolling left causes the preceding field in the record to be placed at the leftmost edit area position; scrolling right causes the next field in the record to be placed at the leftmost edit area position.

- PAGE or P - specifies scrolling by one page.
- HALF or H - specifies scrolling by a half page.
- MAX or M - specifies scrolling to the top, bottom, left margin, or right margin, depending upon which scrolling command is used.
- CSR or C - 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.
- DATA or D - for up and down scrolling, specifies scrolling of one line less than is currently visible on the logical screen (PAGE - 1). For left and right scrolling, it specifies scrolling one column less than is currently visible on the logical screen (PAGE - 1).

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 enter

```
COMMAND ==> 3
```

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

Note: When editing with a format, left and right scrolling is controlled by the format. The scroll amount is ignored; the editor scrolls one field width starting from the field boundary currently displayed on the left side of the screen.

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, CSR, or DATA, overtype the first character with "P", "H", "M", "C", or "D", 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 when the first letter uniquely defines the option. For example:

```
REPLACE MEMBERS ==> NO      (YES or NO)
DISPOSITION      ==> KEEP   (KEEP or DELETE)
```

To change these options to YES or DELETE, overtype the first character with "Y" or "D", respectively.

SPLIT SCREEN

Split screen mode allows you to divide the display screen into two logical screens separated by a horizontal split line. 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 member of a partitioned data set, or editing the same sequential data set) will also be in conflict when attempted simultaneously on the split screens. Since only one logical screen is active at a time, you cannot split the screen while executing a foreground compilation or a TSO command.

You can enter split screen mode by moving the cursor to the desired location for the split line 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, only one of the logical screens is considered active at any one time. The location of the cursor identifies which of the two screens is active. Any interrupts, such as the operation of program function keys, are interpreted as having meaning for the active screen. 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.

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. If you have stacked any other commands after "=X", they are ignored.

On a 3290 terminal, in addition to splitting the screen horizontally, you can issue the SPLITV command to split the screen vertically, for a total of four logical screens.


```

BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- LINE 000000 COL 001 080
COMMAND ==>                                SCROLL ==> PAGE
***** TOP OF DATA *****
COINS:                                00010001
  PROCEDURE OPTIONS (MAIN);           00020000
  DECLARE                              00030000
. . . . .
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> _
                                USERID - BECKT
0 ISPF PARMS - Specify terminal and user parameters  TIME - 12:47
1 BROWSE - Display source data or output listings  TERMINAL - 3278
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 TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES- Perform library management utility functions
C CHANGES - Display summary of changes for this release
T TUTORIAL - Display information about ISPF/PDF
X EXIT - Terminate ISPF using list/log defaults

```

Enter END command to terminate ISPF.

Figure 7. Split Screen Example

Note: The SPLITV function is not active if the data being displayed on a screen is more than 80 characters wide.

HELP INFORMATION

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

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. For some messages in the LMF utilities, entering the HELP command again displays a message help panel, which provides a description of the error condition that would cause the message and a description of the user action needed to correct the error. 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) that lets you specify the log and list processing.
- Enter option X to terminate PDF with your defaults for processing 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."

The termination panel allows you to specify whether the log and list files are to be printed with a background job or routed to a local printer, or deleted or kept without printing. If the files are to be printed, you must specify the destination (SYSOUT class or printer id), as well as job statement information for background printing.

Once you have typed the information, press ENTER to complete termination. The screen is cleared, one or more termination messages are displayed, and you are returned to TSO.

The termination panel shown in Figure 8 is the version that is displayed if both log and list files have been used. If only one has been used, a version of the panel is displayed for only that one. If neither list nor log has been used, PDF terminates immediately when you use either option X or the END or RETURN command from the primary option menu.

SPECIFY DISPOSITION OF LOG AND LIST DATA SETS -----
COMMAND ==>

LOG OPTIONS FOR THIS SESSION

Process option ==> PD
SYSOUT class ==> A
Local printer ID ==>

LIST OPTIONS FOR THIS SESSION

Process option ==> PD
SYSOUT Class ==> A
Local printer ID ==>

VALID PROCESS OPTIONS:

PD - Print data set and delete K - Keep data set (without printing)
D - Delete data set (without printing)

INSTRUCTIONS:

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

JOB STATEMENT INFORMATION: (Required for system printer)
==> //FRANKEL JOB (DC19,B213,896754),'LAGRANDE.....72745'
==>
==>
==>

Figure 8. Termination Panel

CHAPTER 3. LIBRARIES AND DATA SETS

This chapter provides the information you need to handle programming libraries, data sets, 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 (ASM), COBOL source (COBOL), object modules (OBJ), load modules (LOAD), and SCRIPT documentation (SCRIPT).

In the TSO environment, each library is a cataloged partitioned data set, with a 3-level name in the form:

'project-name.group-name.type'

where

- "project-name" is the common identifier for all libraries belonging to the same project.
- "group-name" identifies the particular group of libraries, such as MASTER, TEST, or FRANK.
- "type" identifies the type of information in the library, such as ASM, COBOL, OBJ, LOAD, or SCRIPT. Except for OBJ and LOAD, the type qualifier need not conform to the standard TSO naming conventions.

Most projects use a hierarchy of related libraries to maintain effective 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. This library control can also be provided using the Library Management Facility controls under LM UTILITIES (Option 8). Libraries can be accessed from dialogs using library access services, documented in ISPF/PDF for MVS Services.

PDF allows concatenation of up to four libraries during browsing, source editing, compilation, assembly, link editing, or SCRIPT/VS processing.

ASM	COBOL	OBJ	LOAD	SCRIPT
MEMBER A	MEMBER A	MEMBER D	MEMBER IJK	MEMBER D
MEMBER B	MEMBER D	MEMBER F	MEMBER PQRS	MEMBER Q
MEMBER C	MEMBER E	•	•	MEMBER XYZ
•	MEMBER G	•	•	•
•	•	•	•	•
•	•	•	•	•

Figure 9. ISPF Library Organization

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 FRANK, 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 or the LMPROM service if the higher-level library is controlled.

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) or link edit (load module) is stored in the lowest-level OBJ or LOAD library (the first library in the concatenation sequence).

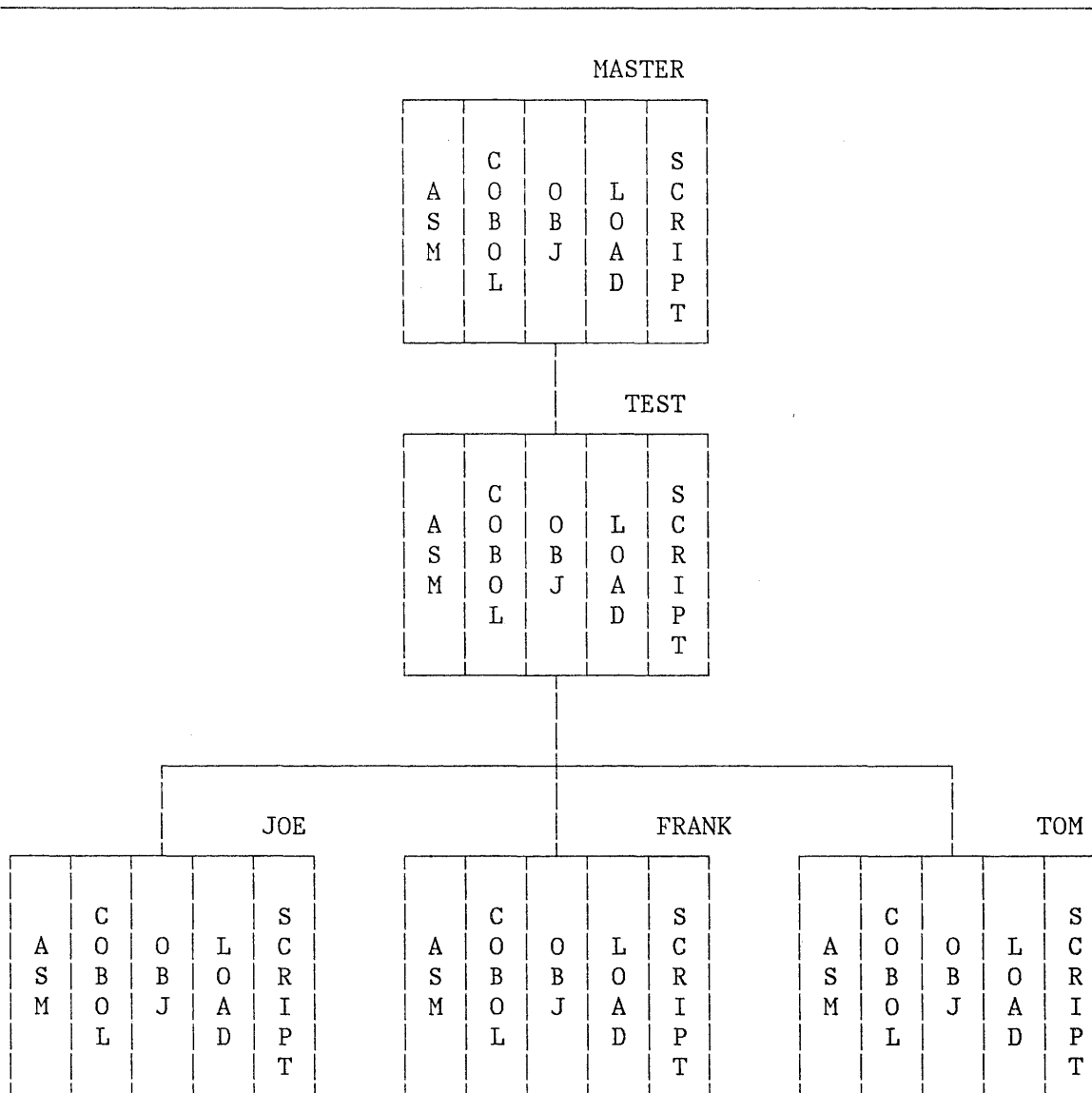


Figure 10. Hierarchy of ISPF Libraries

LIBRARY STATISTICS

At your option, the PDF editor automatically generates and maintains the following statistics for each member of an ISPF library or other partitioned data set:

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, in the national format

Date/Time Modified: When this version was last modified, with the date in the national format

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) or with the LEVEL and VERSION edit primary commands. 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 (for example, "VER.MOD 02.15" means version 2, modification 15).
- For creation date and date last modified, the date is in the national format: for example, "83/06/27" means June 27, 1983; "06/27/83" means June 27, 1983; "27/06/83" means June 27, 1983.
- For time last modified: HH:MM (for example, "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, column headings are displayed in the national language. The following are the column headings displayed:

NAME	- Name of the Member
LIB	- Library Number (See below)
VV.MM	- 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 (group 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. In the member list, the date created and date modified is displayed in the national language format. For example: yy/mm/dd, mm/dd/yy, dd/yy/mm.

```

BROWSE - SPFDEMO.MYLIB.PLI -----
COMMAND ==> _
NAME          VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD   ID
ACCOUNT      01.00   83/01/09   83/01/09 17:07    21    21    0   HUNTER
ACCT1        01.01   83/02/11   83/04/23 14:52   199   193    0   BECKETT
ACCT2        01.00   83/03/09   83/03/09 17:07    20    20    0   BECKETT
COINS        01.04   83/04/24   83/04/28 16:20    19    19    4   JOSLIN
COMPX        01.00   83/01/21   83/01/21 11:08    44    44    0   JOSLIN
COMPY        01.01   83/01/14   83/01/16 12:30    13    13    1   JOSLIN
DCLS         01.00   83/04/23   83/04/23 15:14    20    20    0   LEESB
LISTNEW      01.02   83/04/23   83/05/06 10:00    17    13    6   LEESB
MAIN         01.00   83/01/09   83/01/09 17:08     4     4    0   KIRK
TESTDIR      01.02   83/04/23   83/05/06 09:04    30    43   10   LAIDLAW
UPDATE       01.00   83/03/26   83/04/01 13:08    13    13    0   COONS
**END**

```

Figure 11. Member List Display


```

BROWSE - ISPFPROJ.MYLIB.LOAD -----
COMMAND ==>
NAME          SIZE  TTR  ALIAS-OF AC ----- ATTRIBUTES -----
ISPF          0003A8 00000C          00 FO          RN RU
ISRBRO        0008D8 00001B          00 FO          RN RU
ISRECM        0009A8 00001B          00 FO          RF RN RU   TS
ISRECMBR      000138 000024          00 FO          RN RU
ISREDIT       000C60 000101          00 FO          RN RU
ISRFOR        001230 00010C          00 FO          RF RN RU   TS
ISRFPR        000F48 000117          00 FO          RN RU
ISRFPT        000278 000201          00 FO          RN RU
ISRJB1        000B40 000208          00 FO          RN RU
ISRJB2        001088 000210          00 FO          RN RU
ISRJOB        000E40 00021A          00 FO          RF RN RU   TS
LMOBAD        0003A8 00000C ???????? ?? FO          RN RU
**END**

```

Figure 12. Load Library Member List Display

Load Module Libraries

For load libraries, the directory information displayed in a member list has a different format as shown in Figure 12. The following column headings are used:

NAME	- Name of the Member
SIZE	- Size of the Member
TTR	- The Relative Block Address
ALIAS-OF	- The Name of the Member for Which This Member is an Alias
AC	- The Authorization Code
ATTRIBUTES	- The Member's Attributes as follows:
	FO - Process only by F Level Linkage Editor
	NX - Not Executable
	OL - Only Loadable
	OV - In Overlay Structure
	RF - Refreshable
	RN - Reenterable
	RU - Reusable
	SC - Scatter Format
	TS - Module to be Tested

Question marks (???) are placed in the member list of a load module library for members that have invalid load module directory fields. For example; member LMOBAD, shown in Figure 12, contains invalid ALIAS and Authorization Code information.

PARAMETER SPECIFICATION

Several entry panels require specification of the following:

- Library and data set names
- Data set password (if applicable)
- Job statement information (for submitting background jobs).

These items are described in the following paragraphs.

Library and Data Set Names

To specify a member of an ISPF library, you must enter a project name, group name, type qualifier, and member name. Each of these items may contain up to eight alphanumeric characters, the first character of which must be alphabetic or the special characters '\$', '#', or '@'.

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

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

For convenience, any cataloged data set (sequential or partitioned) with a 3-level name may be entered in this manner, even if the three components of the data set name do not actually identify project, group, and type. For partitioned data sets, if the member name is not specified, a member list is displayed.

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

```
ISPF LIBRARY:
  PROJECT ==> SAMPLE
  GROUP   ==> MYOWN   ==> TEST   ==> MASTER ==>
  TYPE    ==> COBOL
  MEMBER  ==>          (blank for member selection list)
```

In this example, three libraries would be concatenated in the following order:

```
SAMPLE.MYOWN.COBO
SAMPLE.TEST.COBO
SAMPLE.MASTER.COBO
```

You must ensure that the concatenated libraries have consistent record formats, logical record lengths, and block sizes.

Partitioned and sequential data sets may also be specified using standard TSO syntax, as follows:

OTHER PARTITIONED OR SEQUENTIAL DATA SET:

DATA SET NAME ==>

VOLUME SERIAL ==> (If not cataloged)

Any fully qualified data set name may be entered, and it must be enclosed within apostrophes. If the apostrophes are omitted, your TSO user prefix is automatically left-appended to the data set name. For partitioned data sets, a member name enclosed in parentheses may follow the data set name. For example:

'SYS1.PROCLIB(ASMHC)'

If the member name (and parentheses) are omitted, a member list will be displayed.

If both a library name and an "other" data set name are specified on the same panel, the "other" data set name is used. Therefore, to specify a library, the "other" data set name must be blank.

The volume serial parameter may specify a real DASD volume or a virtual volume residing on an IBM 3850 Mass Storage System. To access 3850 virtual volumes, you must also have MOUNT authority (see TSO ACCOUNT command).

Note: PDF does not allow the use of multivolume data sets.

Data Set Password

PDF allows you to protect your data sets using OS password protection. Any data set may be protected to permit read-only or read/write access. More than one password may be assigned to the same data set. A data set that is read/write protected, for example, might allow several authorized users to read it, but only one user to write.

You can specify a data set password in the following manner:

DATA SET PASSWORD ==> (If password protected)

A nondisplay input field is used so that the password does not appear on the screen. For entry panels that allow specification of a concatenated sequence of libraries, the password applies to all data sets in the concatenation sequence.

When you replace a longer password with a shorter password, you must blank out the remaining spaces of the password field.

Note: PDF may be used with either the TSO/VS2 Programming Control Facility (PCF) or the Resource Access Control Facility (RACF). PCF and RACF provide extensive facilities for data set security. With either PCF or RACF, you should not attempt to

enter a password on the PDF panels, since both of these facilities rely on your TSO user id and (logon) password to identify you and check for proper authorization.

Job Statement Information

PDF allows you to submit batch jobs for printing and language processing. Before submitting a batch job, you must supply a job statement. For this purpose, four lines are provided on each job submission panel. The first time a job submission panel is displayed, the job statement information appears as follows:

```
JOB STATEMENT INFORMATION:
====> //user-ida JOB (ACCOUNT),'NAME'
====> //*
====> //*
====> /*
```

The pre-entered job name consists of your TSO user id with the letter "A" right-appended. The last character of the job name is automatically incremented to "B", "C", etc., each time the job statement information is used. You may change the last character of the job name to a numeric digit, rather than a letter, in which case the last character is cycled from 0 to 9, rather than A to Z.

You may overtype the entire job name, if desired, but automatic incrementing of the last character is suspended unless the job name starts with your user id.

You must enter the remaining job statement information the first time you submit a batch job. You can use the lines containing "/*" as continuation lines (by changing the "/*" to "//"), or to enter other JCL statements, such as a JOBLIB DD statement. If you don't need these lines, you may blank them out. Blank lines are not submitted to the job stream.

USER PROFILES

PDF stores information on your behalf in your user profile. This allows PDF to prefill panel input fields with the values that 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), and type
- Job statement information
- SYSOUT class for printed output
- Defaults for list/log allocation and processing

- Terminal characteristics and PF key definitions
- Edit profiles, including mask, tabs, and bounds
- Current scroll amount (browse, edit, and member lists)
- Processing options for each of the language processors
- Data set allocation/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 is requested. But from that point on, you simply verify the information and make any desired changes before proceeding.

Notes:

1. Seven sets of job statements are retained, one set each for the foreground print option (excluding SCRIPT/VS), batch option, hardcopy utility, outlist utility, SCRIPT/VS draft output, SCRIPT/VS final output, and termination. This allows you to provide different job statement parameters for each of these functions.
2. When you replace a longer password with a shorter password, you must blank out the remaining spaces of the password field.

MEMBER LIST COMMANDS

For any of the following PDF options, a member list is displayed if you specify an ISPF library or other partitioned data set but supply no member name:

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

An example of a member list display is shown in Figure 11 on page 29.

Locating Members

You can scroll up and down the member list using 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.

Note: If you enter more than one "S", the topmost member is selected and the rest are ignored.

- 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 creates 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 you in a recursive invocation of edit, by another user, or, in split screen mode, on another logical screen. To display an up-to-date list, return to the browse or edit entry panel, leave the member name blank again, and press ENTER.

Multiple-Selection Member Lists

For the utility options, you can select multiple members from the list by typing a single character 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 data sets called the list file and the log file, respectively.

When needed, the two data sets are allocated automatically. They are temporary data sets named:

```
user-id.SPFn.LIST
user-id.SPFLOGn.LIST
```

where "n" is a digit from 0 through 9.

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

If you have specified a data set prefix that differs from your user id in your TSO profile, the data set names will begin with your user prefix, followed by your user id.

The list file is used to accumulate PDF-produced listings and screen images. 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 image 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 data sets (and members) that were

modified via edit or utility options, batch jobs that were submitted, dialog test trace data, etc.

Appendix B describes the format of the ISPF log and list files.

When you terminate ISPF, you may print the contents of these files. You may direct the printed output to a system output device via a batch job, or you may direct it to a local IBM 3284, 3286, 3287, 3288, or 3289 printer. For local printing, the TSO Command Processor "DSPRINT" must be installed. If the log/list files are printed, they are automatically deleted following printing, and new log/list files are allocated the next time you invoke PDF. If you elect not to print the log or list files, you may delete them, or you may keep them and they will be opened with a disposition of MOD the next time you invoke PDF.

Notes:

1. You may avoid allocation of the list file simply by not requesting any print functions. You may prevent allocation and use of the log file with the Parms option (option 0.2). See Chapter 4.
2. If you use the dialog test option, it is recommended that you allow allocation of the log file.

FOREGROUND AND BATCH OUTPUT LISTINGS

The following additional listing files are allocated as needed for foreground or batch processing:

```
prefix.list-id.LIST  
prefix.list-id.LINKLIST  
prefix.list-id.TERM  
prefix.list-id.TESTLIST
```

where "prefix" is your data set prefix in your TSO profile. The particular file names used depend on the foreground or batch processing option chosen.

For batch processing, the output may either be directed to a list file or printed as part of the batch job. Upon completion of batch processing, you may browse the list file and then invoke the "hardcopy" utility option to print it. Using this utility, you indicate whether the file is to be kept or deleted following printing. PDF does not delete these files during termination.

For the foreground option, the output listing is directed to a list file and automatically displayed (for browsing). Upon completion of browse, a selection panel is displayed that allows you to print, keep, or delete the list file. Again, PDF does not delete this file during termination.

OTHER TEMPORARY FILES

PDF allocates temporary control and listing files, as needed, for its own internal use. They are assigned the names:

```
user-id.SPFTEMPn.CNTL
user-id.SPFTEMPn.LIST
user-id.ISR000z.BACKUP
user-id.SPFxxx.OUTLIST
```

where "n" is the number 1 to 4, depending on which logical screen is active, "z" is a number from 1 to 8, controlled by the edit recovery table (ISREDRT) and the number of concurrent edit invocations active, and "xxx" is a PDF-generated number.

If you have specified a data set prefix different from the user id in your TSO profile, the data set names start with your data set prefix, followed by your user id.

These files are automatically deleted during normal ISPF termination. You are usually not aware of their existence.

CHAPTER 4. ISPF PARMS (OPTION 0)

The Parms option allows you to display and change selected ISPF parameters at any time during the ISPF/PDF session. Changes remain in effect until you change the parameters again, and are saved from session to session. The parameter selection panel is shown in Figure 13.

```
----- ISPF PARAMETER OPTIONS -----  
OPTION ==> _  
  
1  TERMINAL  - Specify terminal characteristics  
2  LOG/LIST  - Specify ISPF log and list defaults  
3  PF KEYS   - Specify PF keys for 3278 terminal with 24 PF keys  
4  DISPLAY   - Specify screen display characteristics  
5  LIST      - Specify list data set characteristics
```

Figure 13. 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 or 3290, and the command stacking delimiter. The panel, with the initial user defaults, is shown in Figure 14. (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 - 3276/3278/3279/3290 terminal)
                  (3278A - 3276/3278/3279/3290 APL keyboard)
                  (3278T - 3276/3278/3279/3290 TEXT keyboard)

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

INPUT FIELD PAD  ==> N     (N - Nulls) (B - Blanks) (Special Character-
                           must not be the same as COMMAND DELIMITER)

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

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

  (3290 only)             (PART - Format using hardware partitions.
                           Effective the next ISPF invocation.)
```

Figure 14. 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 C for the APL and TEXT character set descriptions.

Specify Terminal Characteristics (Option 0.1)

Notes:

1. One or more of the 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

A 5550 3270 Kanji Emulation Version 3 terminal has the same character set as a 3278 Katakana terminal. If you are using a 5550 3270 Kanji Emulation Version 3 terminal, specify 3278KN as the terminal type.

2. To avoid problems in finding and changing text strings on a Katakana terminal while caps mode is off, when the data was entered using caps mode on, set the terminal type to 3278KN.

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.

In the following cases, ISPF automatically senses the terminal type and number of PF keys:

- If the screen size is greater than 24 lines (determined when the user logs on), ISPF sets the terminal type to 3278.
- If you press a PF key higher than 12, ISPF sets the terminal type to 3278 and the number of PF keys to 24.

ISPF cannot sense the terminal type or number of PF keys in the following cases:

- If you have switched between a 3277 and 3278 Model 2 (both 24-line terminals).
- If you have switched from a terminal with 24 PF keys to a terminal with 12 PF keys.

In these cases, you must inform ISPF of the terminal type and number of PF keys using option 0.1 or 0.3. Otherwise, an incorrect character set or incorrect PF 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. The pad character specified may be a B (blank), N (null), or any special (non-alphameric) character. The character you choose for the command stacking delimiter must not be the same as the character that you have chosen for input field padding.

Specification of screen format applies to 3278 Model 5 and 3290 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. For a 3290 terminal, you can specify PART to use the maximum available screen size. The effect of PART is as follows: If the 3290 is configured to permit partitions, ISPF formats the screen in one of four ways, depending on the 3290 configuration:

- 31 rows by 80 columns
- 62 rows by 80 columns
- 31 rows by 160 columns
- 62 rows by 160 columns

In addition, the SPLITV command is enabled when the 3290 is configured to permit hardware partitions. If the 3290 is not configured to permit hardware partitions, the default mode depends on the definition of the terminal in your system.

Note: The SPLITV function is not active if the data being displayed on a screen is more than 80 characters wide.

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.

The character you choose for the command stacking delimiter must not be the same as the character that you have chosen for input field padding.

Note: The browse and edit FIND and CHANGE commands will not work with a search argument that contains the command delimiter, even if string delimiters are used.

Specify Log and List Defaults (Option 0.2)

SPECIFY LOG AND LIST DEFAULTS (OPTION 0.2)

When you select this option, a panel (Figure 15) is displayed that allows you to specify default processing for log and list files, lines per page, and allocation parameters. These entries will be used when you terminate ISPF using primary option X or the RETURN command.

```
----- LOG AND LIST DEFAULTS -----  
COMMAND ==>  
  
LOG DATA SET DEFAULT OPTIONS          LIST DATA SET DEFAULT OPTIONS  
-----  
Process option ==> PD_                 Process option ==> PD  
SYSOUT class   ==> A                   SYSOUT class   ==> A  
Local printer ID ==>                   Local printer ID ==>  
Lines per page ==> 60                  Lines per page ==> 60  
Primary pages  ==> 10                  Primary pages  ==> 100  
Secondary pages ==> 10                 Secondary pages ==> 200  
  
VALID PROCESS OPTIONS:  
PD - Print data set and delete         K - Keep data set (without printing)  
                                        D - Delete data set (without printing)  
  
JOB STATEMENT INFORMATION:             (Required for system printer)  
==> //HOSTETLA JOB (U907,B945),'HOSTETLER RS',NOTIFY=HOSTETL  
==>  
==>  
==>
```

Figure 15. Log and List Defaults Panel

The initial defaults are:

	Log File	List File
SYSOUT Class	==> A	==> A
Lines per Page	==> 60	==> 60
Primary Pages	==> 10	==> 100
Secondary Pages	==> 10	==> 200

No defaults are supplied for the other parameters on this panel.

Normal values for lines per page are:

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

Primary/secondary allocation parameters are specified in terms of the anticipated number of pages of printout. These values are automatically converted by ISPF to the appropriate number of blocks prior to allocating space for the log and list files.

If you modify the primary/secondary allocation parameters after the files have been allocated, the new values take effect the next time you enter ISPF. (The list file is allocated the first time you request a print function. The log file is allocated the first time you perform some action that results in a log message, such as saving edited data or submitting a batch job.)

For the log file, you may specify a primary allocation of 0 (zero) to prevent allocation and generation of the log. You can avoid allocating the list file by simply not requesting any print functions.

Notes:

1. If the processing option for the log file is Keep when PDF is terminated, the log file will be opened with a disposition of 'MOD' the next time PDF is invoked.
2. If you change the defaults to prevent allocation of the log file, and the log is already allocated, you must delete the log file when you terminate ISPF to stop generating the log.
3. If you use the dialog test option, it is recommended that you not prevent allocation of the log file.

If you request default processing options for the log and list files, the following rules apply:

- If you specify PRINT, you must also specify SYSOUT class and job statement information. (If you specify option PD for both log and list, you may specify different SYSOUT classes but only one job is submitted for printing both files.)
- If you specify routing to a local printer, you must also specify a printer id and leave the SYSOUT field blank.

If you do not observe these rules, or if you specify no default processing options, primary option X or the RETURN command causes the termination panel to be displayed.

After you review or change the parameters on this panel, enter the END command to return to the previous panel.

Specify Program Function Keys (Option 0.3)

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 Params panel, the panel shown in Figure 16 is displayed.

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

```
----- 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 16. PF Key Definition Panel (12 PF Keys)

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 16 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 17.

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.

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.

Specify Program Function Keys (Option 0.3)

----- 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 ==> CURSOR

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 17 (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 17 (Part 2 of 2). PF Key Definition Panels (24 PF Keys)

Specify Screen Display Characteristics (Option 0.4)

SPECIFY SCREEN DISPLAY CHARACTERISTICS (OPTION 0.4)

When you select this option, a panel is displayed (Figure 18) that allows you to specify PLACEMENT=ASIS or PLACEMENT=BOTTOM for the command line and long message line.

A specification of ASIS causes the screen to be displayed as specified in the panel definitions for the various panels displayed during the session. A specification of BOTTOM causes command and long message lines to be displayed at the bottom of the logical screen for each display during the ISPF/PDF session. Since the command lines on some of your panels may not be moved to the bottom, you may want to keep the command line placement for existing dialogs set to ASIS. For additional information about restrictions on the use of PLACEMENT=BOTTOM, refer to ISPF for MVS Dialog Management Services. When BOTTOM is in effect:

- The command line is moved to the last line of the logical screen.
- Each line that followed the command line is shifted up one line.
- The long message overlays the next to the last line of the logical screen (the line above the new command line location).

```
----- DISPLAY CHARACTERISTICS -----  
COMMAND ==>  
  
COMMAND LINE PLACEMENT ==> ASIS    (ASIS   - Display as shown  
                                     in panel definition.)  
  
                                     (BOTTOM - Display as the last  
                                     line on the screen or  
                                     as the last line above  
                                     the split line.)
```

Figure 18. Screen Display Characteristics

SPECIFY LIST DATA SET CHARACTERISTICS (OPTION 0.5)

The Specify List Data Characteristics option (option 0.5) allows you to specify the characteristics of the records to be contained in the list data set when it is allocated.

When you specify this option, a panel (Figure 19) is displayed that allows you to specify the record format, the logical record length, and the line length to be printed. When the characteristics are reset, their new values take effect at once unless the list data set has already been allocated. In that case, the new values are used in the next ISPF session. These values are saved in your user profile, which ISPF automatically builds and maintains across sessions.

```
----- LIST DATA SET CHARACTERISTICS -----  
COMMAND ==> _  
  
RECORD FORMAT          ==>  
LOGICAL RECORD LENGTH ==>  
LINE LENGTH           ==>
```

Figure 19. List Data Set Characteristics Panel

The record format specifies the format and characteristics of the records in the list data set. The allowable record formats are:

FBA fixed-length records that contain ANSI-defined printer control characters.

Specify List Data Set Characteristics (Option 0.5)

VBA variable-length records that contain ANSI-defined printer control characters.

The default setting for the record format for the list data set is FBA.

The logical record length specifies the length, in bytes, of fixed-length records or the maximum length allowed for variable-length records. The default value for the logical record length for the list data set is 121. This value represents one ANSI-defined control character and 120 bytes of data to be printed.

The line length denotes the length of the logical line to be printed. If the specified line length is greater than that of the logical record, of the list data set, then the data is truncated. The range of allowable lengths is from 80 bytes to 160 bytes. The default value for the line length for the list data set is 120.

The information supplied by the parameters allows for the printing of panels whose line lengths would not otherwise be supported by the available printing facilities.

For example:

1. A panel of 160 bytes in width is to be printed, but printing capabilities allow only 132 bytes.

The following should be specified:

```
line length - 130
LRECL       - 132 (allows for two ANSI-defined control characters)
RECFM      - FBA or VBA
```

The first page of output would contain the first 130 bytes of the panel, and the second page would contain the last 30 bytes. This technique is referred to as the "cut and paste" method of printing.

2. A panel 132 bytes in width is to be printed; line length is supported by printing facilities.

The following should be specified:

```
line length - 132
LRECL       - 133 (allows for one ANSI-defined control character)
RECFM      - FBA or VBA
```

The entire panel would be printed out on one page of output.

3. A panel of 80 bytes in width is to be printed; default values for the LIST parameters are to be used.

The entire panel would be printed out on one page of output as in the previous release of ISPF.

CHAPTER 5. BROWSE (OPTION 1)

The browse option allows you to display source data and listings stored in ISPF libraries or other partitioned or sequential data sets with the following characteristics:

Record Format (RECFM):

- Fixed, variable (non-spanned), or undefined
- Blocked or unblocked
- With or without printer control characters

Logical Record Length (LRECL): Up to 32,766 characters

When you select browse, a panel (Figure 20) is displayed to allow you to specify an ISPF library or other data set, volume serial (if the data set is not cataloged), data set password (if the data set is protected), browse mode (mixed or conventional), and format name (if the data set you specify is a formatted data set).

```

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

ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>      ==>      ==>      ==>
  TYPE    ==>
  MEMBER  ==>      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>
  VOLUME SERIAL ==>      (If not cataloged)

DATA SET PASSWORD ==>      (If password protected)

MIXED MODE      ==> YES      (Specify YES or NO)

FORMAT NAME     ==>

```

Figure 20. Browse - Entry Panel

You can specify a concatenated sequence of ISPF libraries.

For libraries and other partitioned data sets, you can supply the name of the member to be browsed, or leave the member name blank to request a member list from which a member can be selected.

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

You can browse unformatted mixed data which contains both EBCDIC (one-byte) characters and double-byte character set (DBCS, or two-byte) characters. To do this, you must specify mixed mode. A new field is provided on the BROWSE entry panel for specifying the operation mode:

MIXED MODE ==> YES (Specify YES or NO)

where YES indicates mixed mode BROWSE, and
NO indicates conventional BROWSE.

If your terminal does not support DBCS, the operation mode is ignored.

DBCS strings are enclosed with shift-out (X'OE') and shift-in (X'OF') characters in unformatted mixed data. The shift-out character (referred to as 'SO' hereafter) precedes the DBCS character string, and the shift-in character (referred to as 'SI' hereafter) follows the string.

If the browse line contains invalid mixed data, the line is assumed to be EBCDIC only. Examples of invalid mixed data include: unpaired SO and SI characters, invalid DBCS characters between SO and SI characters, or an odd number of bytes between SO and SI characters.

If you invoke BROWSE from the Library Utility (Option 3.1) or the Data Set List Utility (Option 3.4), mixed mode is the assumed operation mode.

If you wish to browse DBCS data as EBCDIC data, you must browse in non-mixed mode. You can do this by operating from a terminal which does not support DBCS, or by specifying "NO" for the mixed mode field on the BROWSE entry panel. In non-mixed mode, SO and SI characters are not treated as special characters; instead, they are treated as nondisplayable characters. Thus, you can browse the data in the conventional way.

You can browse DBCS data in hexadecimal format, just as you would EBCDIC data. For information on specifying a hexadecimal display, see "Hexadecimal Display (HEX Command)" on page 95.

You can browse a sequential data set containing records that consist of subfields for which locations and lengths are fixed throughout the data set. This type of data set is called a **formatted data set**. The formatted data browse is particularly useful when a subfield contains DBCS data without SO and SI characters. You can browse a formatted data set by specifying a format name in the BROWSE entry panel.

A new field is provided in the Browse entry panel for specifying the format to be used in browsing a formatted data set:

FORMAT NAME ==>

The format name, like a member name, can consist of up to eight alphameric characters, the first of which must be alphabetic.

A format definition can include EBCDIC fields, DBCS fields, and mixed fields. If the specified format includes a mixed field definition, and you specify conventional browse (MIXED = NO) as the operation mode, the operation mode is ignored.

The Format Definition utility is provided to support the IBM 5550 terminal using DBCS. For information on defining formats for formatted data sets, see "Format Definition Utility (Option 3.11)" on page 222.

When formatted data is displayed, an attribute character that does not reside in the data set and is not stored back to the data set, precedes each field. Therefore, the column position on the display is different from the column position in the data set.

The allowable maximum length is decreased two bytes per field definition from the standard browse allowable maximum length.

The browse data display is shown in Figure 21. Any nondisplayable characters in the data are changed on the display to either a period or a character that you have specified. Using the DISPLAY command, you can specify whether printer carriage control characters are to be treated as part of the data, and thus displayed.

During browse, 4-way scrolling is available through 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 - SPFDEMO.XXX.COBOL(CBLMAIN) - 01.01 ----- LINE 000000 COL 001 080
COMMAND ==> - SCROLL ==> PAGE
***** TOP OF DATA *****
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. 'F20D1000'.
000300 DATE-COMPILED. OCT. 20, 1983
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 21. Browse - Data Display

Browse provides seven functions (described in the following sections) each of which is controlled by a command that may be entered in the command input field on line 2:

- Locating Lines (LOCATE)
- Identifying Columns (COLS)
- Resetting Columns Line (RESET)
- Finding Character Strings (FIND)
- Controlling the Display (DISPLAY)
- Controlling Hexadecimal Display (HEX)
- Recursive Browsing (BROWSE)

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 enter the END command again to return to the browse entry panel.

When the entry panel is displayed, you may select another data set or member, or enter the END command 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:

```
.aaaaaaa
```

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

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.

Once a label has been defined in this manner, it may be used as an operand in a subsequent LOCATE command. The latest assignment of a label overrides any previous assignments. Several labels may be assigned to the same line. Labels are not retained when you leave the browse option.

IDENTIFYING COLUMNS

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

The COLUMNS command displays a column identification line on the first line of the data area. 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 22. 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 - SPFDemo.XXX.COBOL(CBLMAIN) - 01.01 ----- 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 22. 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 determine 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).

Note: You cannot use FIND to locate a string containing the command delimiter character, even by using string delimiters.

You can define the FIND string to be an EBCDIC, DBCS, or mixed string. If you delimit a DBCS string with SO and SI characters, the SO and SI characters are not used as part of the string. If you specify a mixed string that contains no EBCDIC character, the string is treated as a DBCS string; that is, the SO and SI characters are not used as part of the string.

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.

If you specify a one-byte hexadecimal string as the FIND string and the string is found at the second byte of a DBCS character, hardware sets the cursor to the first byte of the character. If you then request RFIND, the same data will be found again. To find the next occurrence of the string, you must move the cursor to the next character position before requesting RFIND.

If you are browsing a formatted data set, and a FIND string is found, the field containing the found string is highlighted.

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. Use the RFIND PF key to locate the other occurrences.

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 additional parameters that need not concern the browse user.) The full description of the FIND command is given in the section entitled "Finding, Changing, and Excluding Data" on page 106.

CONTROLLING THE DISPLAY (DISPLAY COMMAND)

The DISPLAY command allows you to display carriage control characters and nondisplayable characters.

The DISPLAY command has the following format:

```

DISPLAY [char] [CC ]
DISPL           [NOCC]

```

The operands are optional and can be entered in either order. If only one operand is entered, the other operand retains its current value.

char is the character to be used to represent nondisplayable characters on the display. It may be a single character, or a single character enclosed in quotes (") or apostrophes ('). A blank must be enclosed in quotes or apostrophes.

CC indicates that carriage control characters are to be displayed and considered part of the data.

NOCC indicates that carriage control characters are not to be displayed and are not to be considered part of the data.

Both operands are stored in your user profile, and are in effect whenever you are using browse. You need to reenter the DISPLAY command only if you want to change one of the operands.

The initial settings for display mode are period (.) and NOCC.

The carriage control character status has no effect if the data that you are browsing has no carriage control characters.

The following examples show how to enter the DISPLAY command:

```
displ " " - to display nondisplayable characters as blanks.
display | - to display nondisplayable characters as "|"s.
DISPLAY NOCC - to suppress the display of carriage control characters.
```

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 next two lines show the same data in hexadecimal representation (Figure 23). A separator line is displayed between the representations for ease in reading.

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 "Finding, Changing, and Excluding Data" on page 106.

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 23. 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 edit (option 2).

```

BROWSE - HUNTER.BHEX ----- LINE 000001 COL 001 080
COMMAND ==>                               SCROLL ==> PAGE
:H3.HEXADECIMAL DISPLAY
7CF4CCECCCCDCD4CCEDDCE
A83B8571453941304927318
-----
WHEN BROWSE IS OPERATING IN HEX MODE, THREE LINES ARE DISPLAYED FOR EACH
ECCD4CDDEEC4CE4DDCDCECDC4CD4CCE4DDCC64ECDCC4DCDCE4CDC4CCEDDCECC4CDD4CCCC
685502966250920675913957095085704645B03895503955201950492731854066905138
-----
SOURCE LINE. THE FIRST LINE SHOWS THE DATA IN STANDARD CHARACTER FORM.
EDEDCC4DCDC444ECC4CCDEE4DCDC4ECDEE4ECC4CCEC4CD4EECDCCDC4CCDCCECD4CDDD4
26493503955B0038506992303955028662038504131095023154194038191335906694B
-----
THE FOLLOWING TWO LINES SHOW THE SAME DATA IN HEXADECIMAL
ECC4CDDDDECD4EED4DCDCE4ECDE4ECC4ECDC4CCEC4CD4CCECCCCDCD
385066336695703660395520286603850214504131095085714539413
-----
REPRESENTATION. SEE :FIGREF REFID='BHEX'..
DCDDCECECECD444ECC47CCDCC4DCCCC77CCCE744
95795255313965B002550A697956095694ED2857DBB
-----

```

Figure 23. Browse - Hexadecimal Display, Vertical

RECURSIVE BROWSING (BROWSE COMMAND)

The BROWSE command allows you to browse another member of the same data set, or browse any other data set, without leaving your current browse session.

The BROWSE command has the following format:

```
BROWSE [member]
```

member

is a member of the ISPF library or another partitioned data set that you are currently browsing.

If the member name is not entered, a panel is displayed. On this panel you can specify any data set or member that can be browsed.

Note: The current browse session is resumed when the nested browse session is terminated.

[The body of the page contains extremely faint and illegible text, likely bleed-through from the reverse side of the document. The text is too light to transcribe accurately.]

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 other partitioned or sequential data sets with the following characteristics:

Record Format (RECFM):

- Fixed or variable (non-spanned)
- Blocked or unblocked
- With or without printer control characters

Logical Record Length (LRECL):

- From 10 to 255, inclusive, for fixed-length records
- From 14 to 259, inclusive, for variable-length records

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 the following in mind:

- The editor truncates variable-length lines by removing blanks at the end of each line during a save.
- 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.
- If the record length of an EDIT file is greater than 72 bytes, a Double-Byte Character Set (DBCS, or two-byte) character could go over the display boundary. In this case, PDF attempts to display the other characters by replacing an unpaired DBCS character byte with a shift-out (X'0E', referred to as 'SO' hereafter) character or a shift-in (X'0F', referred to as 'SI' hereafter) character. If the replaced SO or SI character might have been destroyed, the line number of the line is highlighted. If you change the position of the SO and SI characters on the display, or delete the SO and SI characters entirely, the DBCS character on the boundary is destroyed to keep the rest of the data intact.
- Printer control characters, if present, are displayed and are treated as part of the data. ASA control characters are alphameric and may be edited. Machine control characters, however, are invalid display characters and are replaced on the screen with attribute bytes.

- 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 data set or nonexistent member of a partitioned data set for editing, the initial display contains several inserted lines between the "TOP OF DATA" and "BOTTOM OF DATA" message lines (Figure 24). The quote marks on the left of the screen will be filled in with sequence numbers when you enter information on the lines.
- You can prefix any edit command with an ampersand (&) and keep the command displayed on the command line after the command has executed. This technique allows you to repeat similar commands without reentering the data. For example, if you enter

```
====> &CHANGE ALL ABCD 1234
```

the command is displayed after the change has been made, which allows you to then modify the parameters and issue another CHANGE command.

- You may simplify repetitive sets of edit commands by creating **edit macros**. For information about edit macros, refer to "Macros" later in this chapter and to ISPF/PDF for MVS Edit Macros.

```
----- ISPF LIBRARY SERVICES ERROR -----
COMMAND ==> _

*****
*
* ERROR DESCRIPTION:
*
*   LMF TASK NOT STARTED
*   THE LMF STARTED TASK WAS NOT FOUND.
*
*
* INSTRUCTIONS:
*
*   Press the ENTER key to proceed with Edit.
*   Enter END command to terminate Edit.
*
*****
```

Figure 25. Edit - LMF Error

EDIT SELECTION

The Edit entry panel is shown in Figure 26. For edit, you can 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 26 indicates the entries you would make to display the member list for the concatenated libraries SPFDEMO.MYLIB.PLI and SPFDEMO.MASTER.PLI.

```

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

ISPF LIBRARY:
PROJECT ==> SPFDEMO
GROUP   ==> MYLIB   ==> MASTER   ==>
TYPE    ==> PLI
MEMBER  ==> _      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==> (If not cataloged)

DATA SET PASSWORD ==> (If password protected)

PROFILE NAME ==> (Blank defaults to data set type)

INITIAL MACRO ==>

MIXED MODE ==> YES (Specify YES or NO)

FORMAT NAME ==>

```

Figure 26. Edit - Entry Panel

For information about editing libraries that are controlled under Library Management Facilities, see ISPF/PDF for MVS Library Management.

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 27.

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

You may specify a macro to be executed before you begin editing your sequential data set or any member of a partitioned data set. This initial macro allows you to set up a particular editing environment for the edit session you are beginning. If you leave the field blank, the initial macro (if any) from your edit profile (specified with the IMACRO command) is executed. To suppress execution of a profile initial macro, enter NONE in the INITIAL MACRO field.

You can edit unformatted mixed data which contains both EBCDIC (one-byte) characters and Double-Byte Character Set (DBCS, or two-byte) characters. To do this, specify mixed mode. A new field is provided on the Edit entry panel for specifying the operation mode:

MIXED MODE ==> YES (Specify YES or NO)

where YES indicates mixed mode Edit, and
NO indicates conventional Edit.

If your terminal does not support DBCS, the MIXED MODE field is ignored.

DBCS strings are enclosed with shift-out (X'OE') and shift-in (X'OF') characters in unformatted mixed data. The shift-out character (SO) precedes the DBCS character string, and the shift-in character (SI) follows the string.

If the edit line contains invalid mixed data, the line is assumed to be EBCDIC-only. Examples of invalid mixed data include: unpaired SO and SI characters, invalid DBCS characters between SO and SI characters, or an odd number of bytes between SO and SI characters.

If you wish to edit DBCS data as EBCDIC data, you must edit in conventional mode. You can do this by operating from a terminal which does not support DBCS, or by specifying "NO" for the mixed mode field on the Edit entry panel. In conventional mode, SO and SI characters are not treated as special characters; instead, they are treated as nondisplayable characters. Thus, you can edit the data in the conventional way.

You can also edit DBCS data in hexadecimal format, just as you would EBCDIC data. For information on specifying a hexadecimal display, see "Hexadecimal Display (HEX Command)" on page 95. You should **not** edit a record in hexadecimal format, however, when a DBCS string encroaches on the display boundary. If you edit under this condition, the result is unpredictable.

You can edit a sequential data set containing records that consist of subfields for which locations and lengths are fixed throughout the data set. This type of data set is called a **formatted data set**. The formatted data edit is particularly useful when a subfield contains DBCS data without SO and SI characters. You can edit a formatted data set by specifying a format name on the Edit entry panel.

FORMAT NAME ==>

The format name, like a member name, can consist of up to eight alphameric characters, of which the first must be alphabetic.

A format definition can include EBCDIC fields, DBCS fields, and mixed fields. If the specified format includes a mixed field definition, and you specify conventional edit (MIXED MODE = NO) as the operation mode, the MIXED MODE field is ignored.

The Format Specification utility is provided to support the IBM 5550 terminal using DBCS. For information on defining formats for formatted data sets, see "Format Definition Utility (Option 3.11)" on page 222.

When formatted data is displayed, an attribute character that does not reside in the data set and is not stored back to the data set precedes each field. Therefore, the column position on the display is different from the column position in the data set.

The allowable maximum length is decreased two bytes per field definition from the standard allowable maximum length for edit.

You must have previously allocated space for the selected data set, but it may be empty. Selecting an empty sequential data set or a nonexistent member of a partitioned data set allows you to create new source data.

The selected member or sequential data set 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 user region. If insufficient storage is available, your edit session is abnormally terminated.

```

EDIT --- SPFDEMO.MYLIB.PLI -----
COMMAND ==>
NAME      LIB      VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT   1      01.00  83/01/09  83/01/09 17:07   21   21   0  HUNTER
ACCT1     1      01.01  83/02/11  83/04/23 14:52  199  193   0  BECKETT
ACCT2     1      01.00  83/03/09  83/03/09 17:07   20   20   0  BECKETT
ACCT3     2      01.02  83/01/09  83/04/23 15:07   21   21   3  FISHERL
ACCT4     2      01.00  83/04/24  83/04/30 15:04   99   99   0  FISHERL
ACCT5     2      01.01  83/04/13  83/04/23 15:09   22   20   4  FISHERL
COINS     1      01.04  83/04/24  83/04/28 16:20   19   19   4  JOSLIN
COMPX     1      01.00  83/01/21  83/01/21 11:08   44   44   0  JOSLIN
COMPY     1      01.01  83/01/14  83/01/16 12:30   13   13   1  JOSLIN
DCLS      1      01.00  83/04/23  83/04/23 15:14   20   20   0  LEESB
LISTNEW   1      01.02  83/04/23  83/05/06 10:00   17   13   6  LEESB
MAIN      1      01.00  83/01/09  83/01/09 17:08    4    4   0  KIRK
MINUS     2      01.03  82/10/06  83/01/09 08:57   19   19   2  MAURER
PLUS      2      01.00  83/01/09  83/02/13 10:08   44   44   0  MAURER
TESTDIR   1      01.02  83/04/23  83/05/06 09:04   30   43  10  LAIDLAW
UPDATE    1      01.00  83/03/26  83/04/01 13:08   13   13   0  COONS
ZCOMP     2      01.01  83/01/14  83/03/13 14:30   13   13   1  STALEY
***END***

```

Figure 27. Edit - Member List

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 data field (Figure 28). 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(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     DO 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 28. 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 or macro 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
- Combine several members into one (merge) or split a member into two members (segment)
- Submit data to the job stream
- Save the edited data or cancel without saving
- Sort data
- Delete lines
- Access dialog element models

You can use line commands to:

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

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

Edit Modes, Options, and Profiles

When you are editing, the edit session is controlled by an environment defined by modes and options. Taken together, these modes and options are called the edit profile.

An example of an edit mode is the caps mode. If you are editing a COBOL program, which is in uppercase, and you want all of your input to be translated to uppercase, you can set caps mode on by entering the CAPS ON command. On the other hand, if you are editing text, which is in mixed case, you can enter the CAPS OFF command to set caps mode off.

The modes, which are part of the edit profile and which you control by entering commands, are listed below. This list will give you an idea of the kinds of things that make up your edit environment and which you can control. The items in the list will all be described in detail in later sections.

- AUTOSAVE - to save data automatically when you end
- AUTONUM - to renumber data automatically
- AUTOLIST - to list data automatically
- CAPS - to control translation to uppercase
- HEX - to control the display format for hexadecimal data
- NOTE - to control notes generated by the MODEL command
- NULLS - to allow you to insert data at the terminal
- NUMBER - to control sequence numbering
- PACK - to control the format of data that is saved
- RECOVERY - to control the generation of a recovery data set
- STATS - to control the generation of ISPF statistics
- TABS - to control hardware and software tabbing

In addition to the modes listed above, there are options associated with several of the modes that are also part of the edit profile.

- AUTOSAVE - PROMPT or NOPROMPT options
- HEX - VERTICAL or DATA display options
- NULLS - STANDARD or ALL display options
- NUMBER - STANDARD, COBOL, and DISPLAY options
- TABS - STANDARD or ANY options or a special tab character

Finally there are three special lines that are part of the edit profile. They are:

- BOUNDS - used to specify left and right boundaries that are used by other commands
- MASK - used when creating new lines
- TABS - used to define tab positions

Edit saves the environment, the edit profile, each time that it is changed. When you start an edit session, the profile is retrieved and used to set up the environment. For example, if you have caps mode on at the end of one edit session, it will be set on at the beginning of the next session that uses the same profile.

You may have up to 25 different profiles for different kinds of data. With this capability, you can set up a profile that will be used when you edit COBOL programs, and a different profile to be used when you edit memos. A third profile could be set up for editing test data, and a fourth for editing JCL statements.

Selecting a Profile

You control the profile that is being used for a particular edit session by putting your data into different libraries and letting edit select the profile based on the library type (or the lowest level qualifier of the data set name if you are editing a sequential data set or other partitioned data set). Or you can explicitly specify the profile to be used on the edit data set panel. And you can switch to another profile while you are in edit by entering the profile command and naming the profile that you want to use.

Setting Up a Profile

You set up a profile by simply entering commands to set the various modes, options, and special lines. Whenever the profile is changed, it is also saved, and the next time that you edit data using the profile, it is retrieved and the environment is set up again. This is much easier than it sounds. First, there are defaults for all of the modes, and in most cases, the defaults do not need to be changed. Second, if you decide that you want to change the edit profile, you simply enter the appropriate command. The edit profile is automatically changed, and is automatically saved for you.

Locking Your Profile

Once you have set up a profile exactly the way that you want it, you can lock it. To do this, enter the PROFILE command with the LOCK option. The profile is saved with all of the current modes, options, and special lines, and it is marked so that the saved copy of the profile can not be overwritten. Each time that you begin an edit session, the profile that you begin with will be exactly the way that you locked it. You can change a mode during an edit session, but as long as the profile is locked, the change will affect only the current session, and will not affect the next session.

Suppose that you almost always want caps mode on. Enter CAPS ON to set the caps mode, and then enter PROFILE LOCK. You can change caps mode during an edit session if you need to enter lowercase data, but because the profile is locked, the edit profile containing CAPS OFF will not be

saved at the end of the edit session. The next time that you begin an edit session, the profile will be retrieved, and caps mode will be on.

Note: If you have locked your current profile, you may not change the initial macro name with the IMACRO command.

Special Data Modes

There are four profile modes that fall into a special category because they can be changed by the editor. The four modes, and the reason that the editor might change them are:

- Caps Mode - set on if the editor detects that a member that is to be edited contains no lowercase characters. It will be set off if the editor detects that the member did contain lowercase characters.
- Number Mode - set on, and number options changed if edit discovers that the data contains valid sequence numbers. It will set number mode off if the data does not contain valid sequence numbers.
- Pack Mode - set on if the data being edited has previously been saved in packed format. It will set pack mode off if the data has not been previously saved in packed format.
- Stats Mode - set on if the member being edited currently has ISPF statistics. It will set stats mode off if the member did not previously have ISPF statistics.

Edit changes the special data modes, even if the profile that was used in setting up the environment is locked. This is handled by first reading the data and profile into memory, and then modifying the edit profile (you can think of it as the current profile), without affecting the copy of the profile that is on disk.

When edit changes your profile to correspond to the data, it inserts special warning lines that are displayed as part of the data. If you want to override the change, you simply enter the appropriate command. For example, if edit changes CAPS mode from ON to OFF, because it finds lowercase characters in the data, you simply enter CAPS ON to reset it back on.

Edit users generally want the modes to change automatically to correspond to the data. This allows you to have a single data set, and to use the default profile, even though some members may contain programs (CAPS ON) and other members contain text (CAPS OFF). Some of the members may have statistics that will be maintained, while other members may be stored without statistics. Some members may be in packed data format and other in standard data format. And finally, and perhaps most important, some members may be sequence numbered, and others may not be numbered.

If you have special requirements, you may not want edit to change the special modes automatically to conform to the data. You may want to

have caps mode on, even if the data contains lower case data. Or you may want to generate statistics on output, whether or not the member originally had statistics.

Specifying the CAPS, NUMBER, PACK, or STATS command in an initial macro, specified in the profile, allows you to specify, for a particular profile, how edit is to process these special modes.

Packing Data

Data can be saved on disk in either packed or standard format. The packed data format has the advantage of saving space on disk at the expense of additional processing when the data is read from or written to disk.

The packed data format allows you to edit packed data in a transparent way. Data that is stored on disk in packed format will be read in by edit, and can be edited in the normal way. When the edit session is ended, the data is packed and stored on disk.

The format in which data is stored can be controlled, either from the edit profile or with the PACK command.

Macros

Macros are similar to edit primary commands. The name of the macro, along with any operands, are entered on the command line.

There are no edit macros distributed with ISPF/PDF. All the commands described in this manual are built-in commands. You may have available, at your installation, common macros that have been written and documented for your use. Or you may write your own edit macros.

The rules for entering a specific macro, along with the expected results, are dependent on the particular macro, and should be documented by your installation. If you want to write your own macros, you should read ISPF/PDF for MVS Edit Macros.

Note: The effect of executing a macro will depend on the implementation of the macro, and results such as cursor positioning, output messages, and so on, may or may not conform to the results that you expect from built-in edit commands.

Labels and Line Ranges

When you are editing data, each line of the data is displayed with a number on the left side of the screen. You may assign a label to a line by overtyping the number with the label. The label always starts with a period (".") followed by one to five alphabetic characters. You may not define a label with Z as the first alphabetic character; labels starting with Z are reserved for edit use. Once you have assigned a label to a

line, it is displayed in place of the number whenever the line is being displayed. You may not type a label on a non-data line or the line displayed indicating one or more lines are excluded (not displayed).

The purpose of labels is to allow you to specify a line or a range of lines in a primary command. There is one command, the LOCATE command, in which you can specify a single label. There are a number of other commands in which you can specify two labels to define a range of lines that are to be processed by the command.

A single label is used in the LOCATE command to locate (scroll to) the line with the label. Most commands, however, require two labels to specify a range of lines to be examined or processed by the command.

In the command descriptions, the range, which is always optional, is specified as [range]. When you enter a command, enter two labels to specify the range.

If you do not specify a range, the entire member or file that you are editing is processed.

An example of a command description is:

```
SUBMIT [range]
SUB
```

Examples of using the command are:

```
SUBMIT           - Submit all lines
SUBMIT .s .e     - Submit lines from .S to .E
SUBMIT .s        - INVALID, a range must include two labels
```

For example, the command:

```
CHANGE ALL 'TEST' 'FINAL'
```

can be interpreted to mean: Change all occurrences of "TEST" to "FINAL". Because no explicit range is specified, the CHANGE command starts at the first line of the data being edited, and scans all lines up to and including the last line, changing all occurrences of "TEST" to "FINAL".

The command:

```
CHANGE .A .B ALL 'TEST' 'FINAL'
```

can be interpreted to mean: From the .A line to the .B line, change all occurrences of "TEST" to "FINAL". The change command in this case will start at the .A line and stop when it has processed the .B line.

When labels are used to specify a range, they are always given in pairs that define the first and last lines, inclusively. To process a single line, repeat the label.

```
CHANGE ALL " " " " _ .A .A
```

can be interpreted to mean: Change all blanks to underscores on the .A line.

The commands on which a range can be specified are:

```
CHANGE
EXCLUDE
FIND
LOCATE
RESET
SORT
SUBMIT
```

To eliminate a label, blank it out, or use the RESET command with the LABELS option to eliminate all labels.

Line Ranges

Commands may operate on a single line, a group of lines, or the complete file.

The following edit primary commands normally operate on the entire file that you are editing:

- DELETE - scans the entire file for excluded or non-excluded lines
- RESET - resets the entire file
- SORT - sorts the entire file, or sorts all excluded or non-excluded lines
- SUBMIT - submits the entire file

Other commands optionally start at the first line of the file and may scan through the entire file. These commands are:

- CHANGE - if FIRST or ALL is specified
- EXCLUDE - if FIRST or ALL is specified
- FIND - if FIRST or ALL is specified
- LOCATE - if FIRST is specified

You may want to perform one of the above commands, but limit its operation to a specific range of lines. You can do this by assigning labels to the first and last lines to be processed and then by entering the labels as operands on any of the above commands.

The following rules apply:

- Labels must be assigned to the first and last lines of a range.
- Both the starting and ending labels must be specified on the above commands to identify the range of lines.
- If you want the range of lines to consist of a single line, enter the same label twice.
- As long as both labels are given, their order is not important. The line closest to the first line of the file will be assumed to be the first line of the range, and the line closest to the last line of the file will be assumed to be the last line of the range.

The following special built-in labels can always be used:

- .ZF - first line of the file
- .ZFIRST - first line of the file
- .ZL - last line of the file
- .ZLAST - last line of the file
- .ZCSR - cursor line

The following examples illustrate the use of both user-defined and built-in labels to identify ranges of lines. They show that the order of the labels, and of other operands is not important and that labels as well as keyword operands can be entered in either uppercase or lowercase.

```
loc first chg .start .zcsr
```

locates the first line flagged ==CHG> between the line labeled .START, and the line with the cursor on it.

```
CHG last pre post .HERE .ZFIRST
```

changes the last occurrence of "pre" to "post" between the first line and the line marked with the .HERE label.

```
c pre post all .mylab .zl
```

changes all occurrences of "pre" to "post" from the .MYLAB line to the last line of the file.

```
find higher word .start .end
```

finds word "higher" between the .START line and the .END line.

Note: Specifying a range of lines is a limiting operation. It does not change the way that the command works, but limits the command to the specified lines. For example:

```
find next 'abc' .a .b
```

starts searching at the cursor position, and searches forward, but the lines that are searched are limited to those between, and including, .A and .B.

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 and autosave mode is on:
 - 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, NOTE 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 (or other partitioned data set). If the member was an alias, the alias indicator is turned off.
 - A source listing of the data is recorded in the ISPF list file for eventual printing if autolist mode is on.
2. If autosave mode is off with the PROMPT option, a prompting message is displayed. The user can issue a SAVE command to save the data or a CANCEL command to terminate edit without saving.
3. If autosave mode is off with the NOPROMPT option, data is not saved, as if the user had entered a CANCEL command.
4. 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)
- Controlling the END command (autosave mode)
- Packing data (pack 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)
- Model note display (note 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	Verifies that lines are correctly numbered, and renumbers them to maintain the proper sequence
RENUM	Renums all lines
UNNUM	Turns off numbering

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 partitioned data sets, 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 sequential data set 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 if your display terminal is wide enough to display both data and sequence numbers.

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 one through six and number mode (saved in the edit profile) is "NUMBER ON COBOL." When the data is fetched, column seven will be the first column displayed. If you enter an UNNUM command, the data will be scrolled so that column one is the first column displayed. If you then enter a "NUMBER (or RENUM) COBOL" command, the data will be scrolled back to column seven. But if you enter a "NUMBER (or RENUM) COBOL DISPLAY" command, the data will be scrolled to column one. (The sequence numbers in columns one 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 set "type" (lowest level qualifier in the data set name) 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 empty sequential data sets, the initial setting of number mode is determined by the data in the profile. The default for a new profile is number mode on for standard sequence fields, and for COBOL fields if the data set 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 sequence 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]
NUMB    [OFF]
NUM
```

You can enter the operands in any order. NUMBER ON is the default. The STD, COBOL, and DISPLAY operands are valid 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 in whichever field(s) were 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 when the COBOL operand is coded. When you display a data set with an LRECL of 80 and STD numbering, you will not see the sequence numbers unless you are using a 3278 Model 5 terminal, which displays 132 characters. If you omit the DISPLAY operand, 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, renumbers all lines, starting with number 100 and incrementing by 100, and positions the screen so that column 1 is at the left. 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 at the left. 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:

UNNUMBER
UNNUMB
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 and Control (STATS, VERSION, and LEVEL Commands)

If you are editing a member of a partitioned data set, you can have edit create and then maintain statistics that are saved with the data.

The STATS command controls the generation of statistics for a member of a partitioned data set. If stats mode is on when the member is saved, the statistics are updated (or created if the member did not previously have statistics) and stored in the "user" portion of the directory entry for that member. 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 sequential data sets.

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."

Included in the statistics are version and modification levels. These numbers have no intrinsic meaning, but can be useful in controlling library members.

The VERSION command allows you to control the version number assigned to a member of an ISPF library. The format for the VERSION command is:

```
VERSION  num
VERS
VER
```

num is a number from 1 to 99 that will be assigned as the version number for this member in the statistics.

The LEVEL command allows you to control the modification level that is assigned to a member of an ISPF library.

The format for the LEVEL command is:

```
LEVEL  num
LEV
```

num is a number from 1 to 99 that will be assigned as the modification level for this member in the statistics.

Edit normally increments the modification level the first time that data is changed during an edit session. This incrementing is suppressed if:

- You have set the modification level with a LEVEL command before making the first change.
- Edit has set the modification level to 0 for a new member.

Note: If stats mode is on, and standard sequence number mode is on, the current modification level is stored in the last two positions of the sequence number for any lines that are modified during the edit session, and for any lines that are already marked with a modification level higher than the current modification level. If you enter LEVEL 0 and then save the data, all lines will be reset to level 0.

Controlling the END Command (AUTOSAVE Command)

Autosave mode controls the meaning of the END command:

- If autosave mode is on, data is saved to disk and the edit session is terminated when the END command is entered.
- If autosave mode is off with the PROMPT option, a prompt message is displayed. You can then enter a SAVE command to save data to disk, or you can enter a CANCEL command to terminate edit without saving.

- If autosave mode is off with the NOPROMPT option, data is not saved to disk when the END command is entered. The edit session is terminated just as though a CANCEL command had been entered.

The AUTOSAVE command sets autosave mode on or off and saves it, along with a prompt option, in the edit profile. The AUTOSAVE command has the following format:

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

Operands may be entered in any order.

ON turns autosave mode on

OFF PROMPT turns autosave mode off with the PROMPT option

OFF NOPROMPT turns autosave mode off with the NOPROMPT option

If no operands are specified, ON is assumed. If OFF is specified alone, PROMPT is assumed. If PROMPT is specified alone, OFF is assumed.

Notes:

1. Autosave mode has an effect only if the data is to be saved to disk. If you end an edit session without having changed any data, the edit session is always terminated without saving to disk.
2. Data is considered changed once it has been marked as changed even though you may change it back to its original form. It is also considered changed if you execute a command that could cause the data to be changed, whether or not it is actually changed. For example, shifting a blank line, or changing a word to itself does not actually change data, but they cause the data to be marked as changed.
3. When a SAVE command is invoked, the change status of the data is reset. If you issue SAVE and then END without changing data, the edit session will be terminated without a prompt message since the data is not marked as changed.

Packing Data (PACK Command)

The PACK command sets pack mode on or off and saves it in the edit profile. When pack mode is on, data that is saved will be stored in a packed format. When pack mode is off, data that is saved will be stored in standard format. The PACK command has the following format:

```
PACK  [ON ]
      [OFF]
```

ON causes pack mode to be set on
 OFF causes pack mode to be set off

If no operand is entered, ON is assumed.

The PACK command allows more efficient use of DASD by replacing repeating characters with a sequence indicating the repetition. When the PDF foreground and batch options encounter the SOURCE DATA PACKED field equal to YES, the unpack and expansion is done automatically. Unpredictable results can occur if packed data is used or executed outside of PDF. For example, the LRECL and BLKSIZE of a packed data set can only be changed via PDF option 3. You cannot use the IEBCOPY utility to change the LRECL and BLKSIZE. A packed CLIST will not execute properly because pack mode analysis is not done prior to passing the CLIST to the system.

Automatic Source Listing (AUTOLIST Command)

The AUTOLIST command controls the automatic generation of source listings. If this mode is on when you terminate edit, and the data is changed and will be saved (see AUTOSAVE), the editor records a source listing of the data in the ISPF list data set for eventual printing.

The command has the following format:

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

If you omit the operand, ON is assumed.

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.

When PDF gets data for editing, the setting of caps mode is initialized based on whether or not the data contains lowercase alphabetic characters. If lowercase alphabetic characters are present in the data, caps mode is set to off; otherwise, it is set to on. To override the automatic setting of caps mode during edit initialization, you may include a CAPS command in an initial macro. If the initial caps mode setting differs from the profile setting, a message is displayed indicating that edit has changed the mode. For new members or empty sequential data sets, the initial caps mode setting is unchanged from the profile. The default for a new profile is caps mode on.

Note: The CAPS command does not apply to DBCS fields in formatted data, or to DBCS subfields in MIXED fields. If you specify CAPS, the DBCS fields remain unchanged.

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]
NUL
```

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. Entering ALL causes trailing blanks and all-blank fields to be replaced with nulls. Omitting ALL preserves one trailing blank (followed by nulls); all-blank fields continue to contain blanks.

Blank characters (hexadecimal '40') and null characters (hexadecimal '00') both appear as blanks. Trailing nulls 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.

Tab 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 the ENTER key, 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. You may not use the command delimiter for this 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.

Characters are overlaid with attribute bytes only on the display; the attribute bytes are never recorded in the data. If your data set contains DBCS subfields, however, be aware that attribute bytes can invalidate them. If you activate hardware tabs and cause an attribute byte to be inserted in the middle of a DBCS subfield, it becomes invalid as a DBCS subfield. Instead, it is displayed as an EBCDIC field. When you turn tabs mode off, the attribute bytes are removed and the overlaid characters at each tab position are redisplayed.

Note: When you are in formatted data edit mode, TABS is ignored.

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 the ENTER key. When you press the ENTER key again, the attribute bytes are reinserted.

Profile Display and Control (PROFILE Command)

You may display your current edit profile at any time during your edit session by using the PROFILE primary command. You can also use this command to define a new profile, switch to a different profile, or lock or unlock the current profile. The command has two formats. The first one, for displaying or defining the profile, is:

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

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 8, inclusive, and defines the number of profile lines to be displayed.

If you omit both operands, the contents of the profile are displayed in the data area of the screen (Figure 29). In the figure, the first four 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 as 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 7, that number of lines of the profile is displayed in the order shown in Figure 29. If you type the number 8, 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 8" 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.

To eliminate the profile lines from your display, use the RESET command.


```

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> ....AUTOSAVE ON....AUTONUM ON....AUTOLIST OFF....STATS ON.....
=PROF> ....PROFILE UNLOCK....IMACRO NONE....PACK OFF....NOTE ON.....
=TABS>
=MASK>                                     /*                                     */
=BND$>                                     <                                     >
=COL$>  ---+---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;                             /*                               */

```

Figure 29. Edit - Profile Display

The second form of the PROFILE command has the following format:

```

PROFILE {LOCK }
PROF   {UNLOCK}
PRO
PR

```

LOCK causes the current profile to be saved, and then locked.

UNLOCK causes the current profile to be unlocked.

For information about locking and unlocking the profile, see "Edit Modes, Options, and Profiles" on page 73.

- ON** indicates that data will be displayed in hexadecimal format. If you don't specify ON or OFF, ON is the default.
- OFF** indicates that data will not be displayed in hexadecimal format.
- VERT** causes the hexadecimal representations to be displayed vertically (two rows per byte) under each character. A representation of the HEX VERT display is shown in Figure 23 on page 63. VERT is valid only when hex mode is ON. If you don't code VERT or DATA, VERT is the default.
- DATA** causes the hexadecimal representations to be displayed as a string of hex characters (two per byte), as shown in Figure 30. Since the hex string is twice as long as the data string, it requires two rows. DATA is valid only when hex mode is ON.

The operands may be typed in either order.

Controlling the Edit Boundaries (BOUNDS Command)

The BOUNDS command sets the left and right boundaries and saves them in the edit profile. It provides an alternative to setting the boundaries with the BOUNDS line command (=BNDS>).

The BOUNDS command has the following format:

```
BOUNDS [left-col right-col]
BOUND
BNDS
BND
```

left-col identifies the left boundary column.

right-col identifies the right boundary column.

If the BOUNDS command is entered without operands, the boundaries are set to their default columns. The same column cannot be specified for both boundaries.

Examples of the BOUNDS command are:

```
bnds          - reset the boundaries to the default columns
bounds 1 72 - sets the left boundary to 1, the right boundary to 72
```

Default boundaries are the minimum and maximum columns that can be specified in the BOUNDS command. They include the entire record that is being edited with the exception of any columns being used for sequence numbers. Some examples of default boundaries are:

1 - 80	FIXED LRECL 80 records without sequence numbers
1 - 72	FIXED LRECL 80 records with sequence numbers
9 - 80	VARIABLE LRECL 80 records with sequence numbers
1 -255	VARIABLE LRECL 255 records without sequence numbers

The current boundaries are used by the following commands:

FIND, CHANGE, and EXCLUDE
 TF (Text Flow), TS (Text Split), TE (Text Enter)
 SORT
 Scrolling, left and right
 Shifting line commands
 S (Show) line command

Notes:

1. Setting the boundaries with this command has the same effect as setting the boundaries on the =BNDS> line.
2. If both the BOUNDS primary command and the BOUNDS line command are used in the same interaction, the line command overrides the primary command.

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 set "type" (the lowest level qualifier in the data set name) and whether or not the data is sequence numbered. The default left and right column boundaries are shown in Figure 31.

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

Figure 31. Edit - Default Column Boundary Settings

You can change the boundaries by displaying the boundary definition line and changing its contents, using the BOUNDS line command (see "BOUNDS Line Command"). 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.

Displaying Model Notes (NOTE Command)

When note mode is on, tutorial notes are included when the MODEL command is used. When note mode is off, note lines are excluded when the MODEL command is used. The NOTE command sets note mode on or off and saves it in the edit profile. The NOTE command has the following format:

```
NOTES [ON ]
NOTE  [OFF]
```

ON causes note mode to be set on

OFF causes note mode to be set off

If no operand is entered, ON is assumed.

Note: You can set the note mode the way that you would normally want the MODEL command to work, and then use the NOTE or NONOTE operand on the MODEL command to override the default mode for a particular model.

General Reset (RESET Command)

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

```
RESET [LABEL] [COMMAND] [ERROR] [CHANGE] [SPECIAL] [EXCLUDED] [range]
RES
```

The operands may be entered in any order. If no operands are specified, RESET executes all the options with the exception of LABEL.

LABEL (or **LAB**) indicates that labels should be cleared (undefined).

COMMAND (or **COM**) indicates that pending line commands should be cleared.

ERROR (or **ERR**) indicates that error flags (=="ERR>") should be reset.

CHANGE (or **CHG**) indicates that change flags (=="CHG>") should be reset.

SPECIAL (or **SPE**) indicates that special lines (flagged with (=PROF>, =MASK>, =TABS>, =BNDS>, =MSG>, or =NOTE=) should be deleted.

EXCLUDED (or **X**) indicates that excluded lines should be reset (set non-excluded.)

range specifies two labels. The labels identify the first and last lines of a range of lines. The reset is limited to this range of lines. This operand can be coded with any other operand on the command.

Note: RESET scans every line of data for conditions to be reset. Using 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 an application-dependent data set. Following a system failure, using the same application-ID in which recovery was initiated, you can use the audit trail to replay the edit session.

You can control how many data sets can be recovered with the edit recovery table, a system data set that contains entries for each level of nested editing sessions that can be recovered. For further information about the use of edit recovery tables, see ISPF and ISPF/PDF for MVS Installation and Customization.

When you select edit from the primary option menu, edit displays a special panel indicating that recovery is available (Figure 32). As directed on the panel, you can proceed with recovery, cancel recovery, defer recovery until the next edit session, or enter the END command to return to the primary option menu. If you proceed with, defer, or cancel recovery and you have other data sets that have been recovered, the recovery panel will be redisplayed for each data set.

Note: Recursive Edit cannot be invoked while you are in Edit Recovery.

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

```
RECOVERY  [ON ]
RECOVER   [OFF]
RECOVERY
RECVRY
RECOV
RECVR
REC
```

If you omit the operand, ON is assumed.

Operating with recovery mode off eliminates the I/O operations that maintain the recovery data set 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 data set is being edited.

```

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

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

The following data set was being edited when a system failure or
task abend occurred:

Data set: SPFDemo.SAMPLE.ASM
Volume:  IS423

Instructions:
  Press the ENTER key to continue editing the data set, or
  Enter END command to return to the primary option menu, or
  Enter DEFER command to defer recovery of the specified data set, or
  Enter CANCEL command to cancel recovery of the data set

To continue editing a password protected data set, specify:

DATA SET PASSWORD ==>

```

Figure 32. Edit - Recovery Panel

Macro-Handling Commands

To assist you in using the edit macro facility with the PDF editor, the IMACRO, RMACRO, DEFINE, and BUILTIN commands are provided.

Specifying an Initial Macro (IMACRO Command)

The IMACRO command saves the name of an initial macro in the edit profile. It has the following format:

```

IMACRO  [name]
        [NONE]

```

name is the name of the macro to be executed at the beginning of each edit session.

NONE indicates that no macro is to be executed at the beginning of each edit session.

Examples of the IMACRO command are:

IMACRO STARTUP - to define STARTUP as the initial macro.
IMACRO NONE - to reset the profile with no initial macro.

Once specified, the initial macro is executed at the beginning of each edit session that uses the profile unless it is overridden by an initial macro entered on the edit data set panel. You can enter NONE on the edit data set panel to suppress execution of the initial macro defined in the profile.

Notes:

1. If the current profile is locked, the IMACRO command cannot be executed.
2. An initial macro can be used to set up your edit environment if you want to default to values other than those automatically set up by edit. For example, if you want CAPS ON, regardless of whether data contains lowercase data, create an initial macro with a CAPS ON command. Edit will first read the profile and the data and then set the caps mode to correspond to the data. Finally it will execute the initial macro, which will override the caps mode as set by edit.

Examples of commands that may be useful in an initial macro, either unconditionally or for new members (members with zero lines):

CAPS - force caps mode on or off
NUMBER - force number mode off
PACK - force pack mode on or off
STATS - force stats mode on or off
RESET - reset unwanted information messages
VERSION - set version number
LEVEL - set or increment modification level

Specifying a Recovery Macro (RMACRO Command)

The RMACRO command saves the name of a recovery macro in the edit profile. The recovery macro is executed, like an initial macro, at edit recovery time, after data has been processed by edit recovery, but before being displayed for subsequent editing. The name of the recovery macro is associated with the current edit session. The RMACRO command has the following format:

RMACRO [name]
[NONE]

name is the name of the recovery macro to be executed. The name may be preceded by an exclamation point (!) to indicate that it is a program macro.

NONE indicates that no recovery macro is to be executed.

Examples of the RMACRO command are:

```

RMACRO RESTART - to define RESTART as the recovery macro.
RMACRO NONE    - to reset the profile with no recovery macro.

```

Note: If the current profile is locked, the RMACRO command cannot be executed.

Define a Name (DEFINE Command)

The DEFINE command is used to identify a command name to edit, or to cancel the effect of a previous DEFINE command. A command name is identified as a CLIST or program macro, an alias of another command name, or a NOP.

```

DEFINE name {CMD MACRO }
DEF         {PGM MACRO }
           {ALIAS name2}
           {NOP       }
           {RESET     }

```

name is the user invocation name.

CMD MACRO identifies 'name' as a command language macro, which is to be invoked by the SELECT CMD service. May also be specified as MACRO CMD or MACRO.

PGM MACRO identifies 'name' as a program (load module) macro, which is to be invoked by the SELECT PGM service. May also be specified as MACRO PGM.

ALIAS name2 identifies 'name' as an alias of 'name2' with the same characteristics

NOP identifies 'name' as a NOP. When 'name' is invoked, nothing is executed. Any aliases are set to NOP, also.

RESET resets the most recent definition of 'name' to the status in effect prior to that definition. The reset becomes a NOP if 'name' is currently NOP.

Examples of the DEFINE command are:

```

DEFINE IJKDOIT MACRO      - Define the name IKJDOIT as a CLIST macro.
DEFINE DOIT ALIAS IJKDOIT - Define the name DOIT as an alias of the
                           macro IKJDOIT.
DEFINE SAVE NOP          - Define the name SAVE to have no effect.
DEFINE SETITUP PGM MACRO - Define the name SETITUP to be a program
                           macro.

```

DEFINE FINDIT MACRO CMD - Define the name FINDIT to be a CLIST macro.
 DEFINE SAVE RESET - Reset the definition of the name SAVE.

Execute a Built-in Command (BUILTIN Command)

The BUILTIN command is used to execute a built-in edit command, even though a macro with the same name may have been defined.

For example, a DEFINE MACEND ALIAS END could be issued so that when an END command is processed by edit, the user-defined MACEND macro is executed. In order to execute the built-in END command, you should issue the command BUILTIN END to actually terminate edit.

BUILTIN cmdname

cmdname is the built-in command to be executed.

Examples of the BUILTIN command are:

BUILTIN END - to execute the built-in END command.
 BUILTIN CHANGE ALL " " "-" - to execute the built-in CHANGE command.

Locating Lines (LOCATE Command)

The LOCATE command performs up or down scrolling, as required, to a specified line. There are two forms of the LOCATE command.

The first form is a specific locate, used to position to a particular line. Either a line number or a label may be specified. This form has the following format:

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

label causes a scroll to the line with the specified label. An error message is displayed if the label is not currently assigned.

line-number causes a scroll to the line with the specified line number. If the data is sequence numbered, and the sequence number does not exist, the preceding line is displayed at the top of the data area.

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").

The second form of the LOCATE command is a generic locate. It is used to position to the first, last, next, or previous occurrence of a particular type of line. This form has the following format:

```
LOCATE [NEXT ] [LABEL  ] [range]
LOC   [PREV ] [COMMAND ]
L     [FIRST] [ERROR  ]
      [LAST ] [CHANGE  ]
      [SPECIAL]
      [EXCLUDED]
```

The operands may be entered in any order.

- NEXT to search from the cursor line, proceeding forward.
- PREV to search from the cursor line, proceeding backward.
- FIRST to search from the first line, proceeding forward.
- LAST to search from the last line, proceeding backward.
- LABEL (or LAB) to search for a line with a label.
- COMMAND (or COM) to search for a line with a pending line command.
- ERROR (or ERR) to search for a line with a error flag (==ERR>).
- CHANGE (or CHG) to search for a line with a change flag (==CHG>).
- SPECIAL (or SPE) to search for a special line (with ==PROF>, =MASK>, =TABS>, =BNDS>, ==MSG>, or =NOTE=)
- EXCLUDED (or X) to search for an excluded line.
- range two labels which define the range of lines to be searched.

Examples of the LOCATE command are:

```
LOCATE SPE - Locate the next line with a special label.
L ERR FIRST - Locate the first line labeled with ==ERR>.
LOC NEXT LABEL - Locate the next line with a label.
L X .START .END - Locate the next excluded line between .START and
                  .END
L FIRST .E .S X - Locate the first excluded line between .S and .E
```

Finding, Changing, and Excluding Data

The FIND, CHANGE, and EXCLUDE commands, described in this section, allow you to find a specified character string, to change one character string to another, or to exclude lines on which a specified character string is found. While these commands provide powerful editing functions, they operate on a complete file rather than on a single data or text line. Therefore, you enter them in the command field on line 2 of the display rather than in the line command area.

Since these commands are generally used together (to find a character string, and then change it to another string or exclude it from the display), they are discussed together. The three 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
- Labeling of a defined line range

The following command summary describes the basic forms of FIND, CHANGE, and EXCLUDE, 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, CHANGE, and EXCLUDE 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  
CHA  
C
```

```
EXCLUDE string-1 [ALL]  
EX  
X
```

If you omit the ALL operand, the FIND, CHANGE, or EXCLUDE 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 PF key to find or exclude each successive occurrence of the string specified in the last FIND, CHANGE, or EXCLUDE command. Use the RCHANGE PF key to change successive occurrences of the string specified in the last CHANGE command.

If you specify a one-byte hex string as the FIND string and the string is found at the second byte of a DBCS character, hardware sets the cursor to the first byte of the character. If you then request RFIND, the same data will be found again. To find the next occurrence of the string, you must move the cursor to the next character position before requesting RFIND.

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, CHANGE, or EXCLUDE command searches for all occurrences of string-1, starting at the top of the data, and positions the cursor at the first occurrence. The second-level verification message (issued when you enter the HELP command in response to the short 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
EX ALL QRST
```

to find the next occurrence of XYZ, to change all occurrences of ABC to IJKL, or to exclude all lines containing the string QRST.

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'
EXCLUDE 'no one'
```

to find the next occurrence of "every one", to change the next occurrence of "every one" to "all", or to exclude the next occurrence of 'no one'.

Note: You cannot use FIND to locate a string containing the command delimiter character, even by using string delimiters.

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 overtyping the line number or data on those lines.

If you are editing a data set containing both EBCDIC and DBCS data, you should note the following rules when using the CHANGE command:

- The SO and SI characters that delimit the string are used as part of the string only if necessary. If you specify replacement of an EBCDIC string with a DBCS string, they are used. If you specify replacement of a DBCS string with another DBCS string, they are not used.
- If you specify that an SO or SI character be changed to another character, the result is unpredictable.
- If you specify a string that causes a subfield length of zero and the BNDS boundary falls between the SO and SI characters, the adjacent SO/SI or SI/SO character strings are replaced with a DBCS blank. If the BNDS boundary does not fall between the SO and SI characters, the adjacent SO/SI or SI/SO characters are removed.
- If the lengths of the two strings specified in the CHANGE command are different, the following occurs:
 - If string-1 is shorter than string-2, the data to the right of string-1 is shifted to the left up to some breakpoint. Breakpoints include the border between an EBCDIC subfield and a DBCS subfield, a double or single blank, or the right BNDS boundary.
 - If string-1 is longer than string-2, any blanks in the record to the right of string-1 are used to make room. When blanks in a DBCS subfield are used, they are used in units of two bytes.
- If a DBCS subfield crosses the right BNDS boundary, the CHANGE command might cause an odd-length DBCS subfield. If this happens, the right BNDS boundary is ignored, and the operation takes place.

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 the ENTER key, 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 33 shows a before-and-after example of the CHANGe command.

```

EDIT --- SPFDemo.XXX.COBOL(CBLMAIN) - 01.01 ----- 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 33 (Part 1 of 2). Edit - CHANGE ALL Command Example

```

EDIT --- SPFDEMO.XXX.COBOL(CBLMAIN) - 01.01 ----- 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 33 (Part 2 of 2). Edit - CHANGE ALL Command Example

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.

Detailed Command Description

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

```

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

```

```

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

```

```

EXCLUDE string-1 [range] [NEXT ] [CHARS ] [X ][col-1 [col-2]]
EXCLUDED          [ALL ] [PREFIX] [NX]
EXC               [FIRST] [SUFFIX]
EX                [LAST ] [WORD  ]
X                 [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 and the optional line range parameter.

In the FIND, CHANGE, and EXCLUDE 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. The default is to disregard uppercase and lowercase when searching. Refer to "Use of Text Strings" for additional information. It may also be specified in any one of the following forms. With the exception of the character (C) string, differences between uppercase and lowercase strings are ignored. For example, FIND 'CONDITION NO. 1' would successfully find any of the following:

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

Within the string itself, alphabets may be typed in uppercase or lowercase (even if caps mode is off). For example, all the following have the same effect:

```
FIND 'Edit Commands'    FIND 'EDIT COMMANDS'    FIND 'edit commands'
```

- 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 hexadecimal string: Any delimited string of valid hexadecimal characters, preceded or followed by the character X. Example: X'C27B'
- A character string: Any delimited string of characters, preceded or followed by the character C. Example: C'THIS WORD' See the discussion under "Use of Character 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, CHANGE, or EXCLUDE command to be used again.

To find, change, or exclude lines that begin with a direction keyword, you must include the direction keyword, and begin the string with either apostrophes or quotation marks. For example:

```
EXCLUDE LAST "LAST BUT NOT LEAST"
```

Note: To find a keyword such as NEXT or ALL, place the keyword in quotation marks.

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 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.
- If a picture string is specified for string-2, it must be exactly the same length as string-1 and can only contain the special characters '=', '>', and '<'.

You can define string-1 and string-2 to be EBCDIC, DBCS, and mixed strings in any combination. If you delimit a DBCS search string (string-1) with SO and SI characters, the SO and SI characters are not used as part of the string. If you specify a mixed string that contains no EBCDIC character, the string is treated as a DBCS string; that is, the SO and SI characters are not used as part of the string.

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

```

If you do not specify an operand, the default is CHARS.

If you specify a mixed string or a DBCS string for the FIND command in conjunction with the PREFIX, SUFFIX, or WORD operand, only the following strings are found:

- PREFIX - A DBCS string preceded by an SO character, a DBCS blank character, or an attribute character of formatted data editing.
- SUFFIX - A DBCS string followed by an SI character, a DBCS blank character, or an attribute character of formatted data editing.
- WORD - A DBCS string that satisfies both the PREFIX and the SUFFIX conditions.

Use of Text Strings

Text strings are processed exactly the same as delimited strings. They are provided for compatibility with prior versions of the product.

Use of Character Strings

A character string, which may be used as string-1 in a FIND, CHANGE, or EXCLUDE command, requires that the search be satisfied by an exact character-by-character match. Lowercase alphabetic characters match only with lowercase alphabetic characters and uppercase alphabetic characters match only with uppercase.

For example, FIND C'XYZ' would find the characters XYZ only if they were in uppercase. FIND C'xyz' would find the characters xyz only if they were in lowercase.

If you specify a text string that contains any SO or SI characters, the string is considered a character string.

Use of Picture Strings for String-1

A picture string used as string-1 in a FIND, CHANGE, or EXCLUDE 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:

<u>STRING</u>	<u>MEANING</u>
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 may be used in a picture string:

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.

A DBCS subfield cannot be specified as the subject of a picture string for the FIND operation.

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'0205' FOUND".

Use of Picture Strings for String-2

In a CHANGE command, string-2 may be a picture string with the following rules and restrictions:

- The length of string-2 must be the same as the length of string-1.
- The only valid special characters are '=', '>', and '<':

<u>STRING</u>	<u>MEANING</u>
---------------	----------------

'='	- 'from' character to be unchanged
'>'	- if 'from' character is lowercase alphabetic, it is changed to uppercase; otherwise, it is unchanged
'<'	- if 'from' character is uppercase alphabetic, it is changed to lowercase; otherwise, it is unchanged

Examples of CHANGE commands:

```
CHG P'@@##' P'>>=='
```

changes an alphabetic, alphabetic, numeric, numeric string so that the alphabetic characters become uppercase and the numeric characters are unchanged.

```
CHG ALL P'=' P'>'
```

changes all characters to uppercase.

Excluded Line 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, CHANGE, or EXCLUDE command:

X or EX 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 displayed (i.e., it is "popped out" from the block of excluded lines) when using FIND or CHANGE.

Column Limitations

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).

Range of Lines Limitation

You can limit the lines to be searched by first assigning a label to the first and last lines to be searched, and then specifying the labels on the FIND, CHANGE, or EXCLUDE command. If you want to limit the search to a single line, assign a label to that line, and then specify the label twice, as the first and last line of the range.

Example of Limiting Operands

The command

```
CHG ALL ' ' '-' 1 5 .S .E X
```

changes all blanks to hyphens (-) in columns 1 to 5 between lines labeled .S and .E, but only those lines that are excluded.

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, a line command is 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 to be confused with the line commands of the same name) to specify a member of a partitioned data set or a sequential data set to be copied into the data being edited. If you use the MOVE command, the member or sequential data set 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] [AFTER label]
      [BEFORE label]
```

```
MOVE [member-name] [AFTER label]
      [BEFORE label]
```


If you enter the member-name operand, the member is fetched from the data set (or concatenated data sets) being edited. If you omit the member-name, you must specify a destination with an A or B line command or with the "BEFORE label" or "AFTER label" operand, where "label" is a system- or user-defined label. A panel is then displayed to allow you to specify any partitioned data set member or sequential data set 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 sequential data set). A number may not follow the A or B line command when it is used with a COPY or MOVE primary command.

When copying large data sets, you can reduce the processing time significantly by specifying NUMBER OFF before the copy operation and NUMBER ON after the copy.

The COPY 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 34, Part 1, shows an example of the COPY command. Member CBLCODE1 is being edited in data set 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 34, Part 2, lines 1000 through 2300 of member ENVD in data set 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 34 (Part 1 of 2). Edit - COPY Command Example

```

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

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

FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>
  VOLUME SERIAL ==>           (If not cataloged)

DATA SET PASSWORD ==>           (If password protected)

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

Press the ENTER key to copy.
Enter END command to cancel copy.

```

Figure 34 (Part 2 of 2). Edit - COPY Command Example

Segmenting Data (CREATE and REPLACE Commands)

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

```

CREATE [member-name] [line-range]
CRE

```

```

REPLACE [member-name] [line-range]
REPL
REP

```

If you enter the member-name operand, the member is written in the data set currently being edited. (For a concatenated sequence of libraries, the member is always written in the first library of the concatenation sequence.) If you omit the member-name, you must specify the source

with an M (move) or C (copy) line command or specify a line range with a pair of line labels. Then a panel is displayed to allow you to specify any partitioned data set member or sequential data set 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 35 shows an example of the REPLACE command. Member CBLCODE1 is being edited in data set SPFDEMO.XXX.COBOL. Lines 800 through 1700 are to be moved, as indicated by the "MM" commands. The entire contents of data set TEMP.SAVE are to be replaced with lines 800 through 1700.

Note: The data that is written to the destination data set 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 35 (Part 1 of 2). Edit - REPLACE Command Example

```

----- EDIT - REPLACE -----
COMMAND ==>
"CURRENT" DATA SET: SPFDemo.XXX.COBOL(CBLCODE1)

TO ISPF LIBRARY:
PROJECT ==> SPFDemo
GROUP   ==> XXX
TYPE    ==> COBOL
MEMBER  ==>

TO OTHER SEQUENTIAL DATA SET OR PARTITIONED DATA SET AND MEMBER:
DATA SET NAME ==> TEMP.SAVE
VOLUME SERIAL ==>          (If not cataloged)

DATA SET PASSWORD ==>          (If password protected)

Press the ENTER key to replace.
Enter END command to cancel replace.

```

Figure 35 (Part 2 of 2). Edit - REPLACE Command Example

Deleting Lines (DELETE Command)

The DELETE command removes (deletes) one or more lines from the data being edited.

There are two forms of the DELETE command. The first form removes a specific line or range of lines. This form has the following format:

```

DELETE ALL range
DEL

```

range two labels that define the first and last lines to be deleted.

The second form of the DELETE command removes either excluded or non-excluded lines. This form has the following format:

```

DELETE {ALL X } [range]
DEL    {ALL NX}

```

The operands may be entered in any order.

ALL X indicates that only excluded lines are to be deleted.

ALL NX indicates that only non-excluded lines are to be deleted.

range two labels that define the first and last lines to be examined for excluded or non-excluded lines.

If the range is omitted, the entire file is searched for either excluded or non-excluded lines.

Note: The DELETE command can be used to "clean up" a large file by excluding lines that meet some criterion, or by leaving all lines that meet the criterion non-excluded. Then, with one command you can delete a large number of lines.

For example, to delete all blank lines in a file, execute the following commands:

```
RESET X          - reset all excluded lines
EXCLUDE ALL P'^' - exclude lines containing non-blank characters
DEL ALL NX       - delete the non-excluded (blank) lines
```

or

```
EXCLUDE ALL      - exclude all lines
FIND ALL P"~"    - find all lines containing a non-blank character
DEL ALL X        - delete the remaining excluded (blank) lines
```

The EXCLUDE primary command, used to exclude lines, is described elsewhere in this chapter. The X (Exclude) and S (Show) line commands are described later in this chapter, under "Line Commands."

Sorting Data (SORT Command)

The SORT command orders data within the file that you are editing.

The SORT command operates in two different modes, based on the hexadecimal mode status. If HEX is on, the data is ordered according to its hexadecimal representation. If HEX is off, you are a national language user, and your installation has defined a special collating sequence table, the data is ordered according to the sequence specified in the collating sequence table.

Sorting is limited to data within the current boundaries. Up to five sort fields can be specified by giving starting and ending columns, and each field can be identified as having data sorted in either ascending or descending order.

Optionally, sorting can be limited to a range of lines by specifying the labels of the first and last lines of the range. Sorting can also be limited to excluded lines, or to non-excluded lines.

The simplest form of the SORT command requires no operands.

SORT

The data within the current boundaries is compared character-by-character and then the data is ordered, line-by-line, in ascending EBCDIC collating sequence. Data outside the current boundaries is ignored, both in comparing the data, and in reordering it. This makes it important to properly set the boundaries before issuing the SORT command. It also makes the command a powerful tool in editing data that may be formatted in multiple columns. You can set the boundaries, for example, to the first half of a record, and you can sort one column of data. Then you can set the boundaries to the last half of the record, and you can sort a second column of data.

When sorting data which contains DBCS character strings, you must ensure that no DBCS string crosses the BNDS boundaries. Also, all records must have the same format at the boundaries, although the format of the left and right boundaries may differ.

If a boundary divides a DBCS character, or if all records do not have the same format at the boundaries, the result is unpredictable.

The complete format for the SORT command, showing all optional parameters, is:

```
SORT [range] [X ] [sort-field1 ... sort-field5]
      [NX]
```

range

Two labels that define the first and last lines to be sorted. If you do not specify a range, the entire file will be sorted.

X

Sort only lines that are excluded

NX

Sort only lines that are not excluded

sort-field

Fields to be used in comparing data. You can specify up to five sort-fields as follows:

```
[A] [start-col end-col]
[D]
```

where "A" indicates ascending order, and "D" indicates descending order. The "A" or "D" can either precede or follow the column specification. "A" is assumed if neither "A" nor "D" is specified.

Start-col and end-col define the field that is to be compared. Both must be within the current boundaries.

If several fields are specified, both the starting and ending columns of each field must be specified, and the fields cannot

overlap. If "A" or "D" is specified for one field, it must be specified for all fields.

The following special simple formats can be used:

- SORT** The single sort field is ascending, with the start-column being the left boundary, and the end-column being the right boundary.
- SORT D** The single sort field is descending with the start-column being the left boundary, and the end-column being the right boundary.
- SORT 5** The single sort field is ascending with the start-column being column 5, and the end-column being the right boundary.
- SORT 5 D** The single sort field is descending with the start-column being column 5, and the end-column being the right boundary.

Recursive Editing (EDIT Command)

The EDIT command allows you to edit another member of the same data set, or edit any other data set, without leaving your current edit session.

The EDIT command has the following format:

```
EDIT [member]
```

member

is a member of the ISPF library or another partitioned data set that you are currently editing.

If the member name is not entered, a panel is displayed. On this panel you can specify any data set or member that can be edited.

Note: The current edit session is resumed when the nested edit session is terminated.

Submitting to a Job Stream (SUBMIT Command)

You can use the SUBMIT command to submit the data being edited (the entire member or sequential data set) to the background job stream. The command has the following format:

```
SUBMIT [range]
SUB
```

range The labels of the first and last lines to be submitted.

Job submission is accomplished by invoking the TSO SUBMIT command.

Note: PDF does not supply a job statement when the edit SUBMIT command is invoked. You may supply job statements as part of the data being submitted. When a job statement is supplied, only the job name will be logged to the ISPF log data set to ensure the protection of sensitive data.

Saving and Cancelling

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 the 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 to the same data set from which it was fetched, except when a concatenated sequence of partitioned data sets 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 data set, the entire data set is rewritten. For a partitioned data set, 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 because of 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 data set by taking the following steps:

1. Enter a CREATE or REPLACE command (with no operand) on the primary command line. (Use CREATE only if the destination is a partitioned data set.)
2. Enter "C99999" or "M99999" on the first data line, indicating that all lines are to be copied or moved. Then press the ENTER key.
3. Fill in the data set (and member) name on the CREATE or REPLACE panel, and press the ENTER key.

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, such as compress (for a partitioned data set) or delete and reallocate (for a sequential data set), 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 issuing 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 the ENTER key.

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. Since the line commands are processed from top to bottom in the data set, it is possible to have one error message appear that masks a later error condition. Only the first error condition encountered is displayed. After you have corrected that error condition, processing can continue and the next error condition (if any) can be displayed. 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 overwrite 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)	M (Move)
D (Delete)	C (Copy)
R (Repeat)	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 the ENTER key, 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 36 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 36 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 36 (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
001000
001100
001200       SYSPRINT FILE STREAM OUTPUT PRINT;
001300       DO HALVES = 100 TO 0 BY -50;
001400       DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001500       DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001600       NICKELS = 100 - HALVES - QUARTERS - DIMES;
001700       COUNT = COUNT + 1;
001800     END;
001900   END;
002000 END COINS;
***** ***** BOTTOM OF DATA *****

```

Figure 36 (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 the ENTER key, 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 command, 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 37.

```
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 37 (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 37 (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 the ENTER key, 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 38 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
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;
A_          ELSE
002000              ;
002100          END;
002200          END TESTDIR;
***** ***** BOTTOM OF DATA *****
```

Figure 38 (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 38 (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 Parns 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 the ENTER key, 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.

Note: All nonblank characters of the mask line that are within the currently defined bounds will be set to a blank for each text entry line.

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 the ENTER key.

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.

If you are using the TE line command in a formatted data set, special considerations include:

- The current BNDS boundaries are automatically modified during command execution, and are reset to the original values after execution is complete. Modifications are as follows:
 - If the left boundary falls on the second byte of a DBCS character in a DBCS field, the boundary is shifted to the left by one byte.
 - If the right boundary does not fall on the same field as the left boundary, it is set to point to the last byte of the field in which the left boundary falls. If it falls on the same DBCS field as the left boundary, and it also falls on the first byte of a DBCS character, the right boundary is shifted to the right by one byte.
- The attribute of the field where the left boundary falls is used for the text input area attribute. The new input data is reformatted to fit within the current boundaries.

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 the ENTER key. 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 39 shows a before-and-after example of line splitting. "TS3" was entered on line 60 and the cursor was placed after the word "text".

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.

```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- COLUMNS 001 072
COMMAND ==>                                     SCROLL ==> HALF
000056
000057 :h6.Text Split(TS) Command
000058 :p.
000059 The TS 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 39 (Part 1 of 2). Edit - Text Split Example

```

EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- 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 39 (Part 2 of 2). Edit - Text Split Example

You can split a line within a DBCS string based on the following rules:

- When splitting at a DBCS character, an SI character is added to the end of the line and an SO character is added at the beginning of the new line.
- If the cursor is placed at the SO character, the SO character becomes the first character to be moved.
- If the cursor is placed at the SI character, the character following the SI character becomes the first character to be moved.
- If the MASK contains DBCS subfields and some of the DBCS subfields cross the left, right, or both column boundaries, the result is unpredictable.

If you are using the TS line command in a formatted data set, special considerations include:

- The current BNDS boundaries are automatically modified during command execution, and are reset to the original values after execution is complete. Modifications are as follows:

- If the left boundary falls on the second byte of a DBCS character in a DBCS field, the boundary is shifted to the left by one byte.
- If the right boundary does not fall on the same field as the left boundary, it is set to point to the last byte of the field in which the left boundary falls. If it falls on the same DBCS field as the left boundary, and it also falls on the first byte of a DBCS character, the right boundary is shifted to the right by one byte.

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 it 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 40 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 --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- 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 the line to the first
000035 data position of 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 40 (Part 1 of 2). Edit - Move Sentence Example

```

EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- COLUMNS 001 072
COMMAND ==>                                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 40 (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, or to the last column being displayed, whichever comes first. However, you can "squeeze" the text to a lesser width, or "expand" the text beyond the last column being displayed by entering a number after the TF command. The number specifies the rightmost column position for the restructured text. If a number greater than the right boundary is specified, the right boundary is used.

The use of the rightmost display column as a default column is a change from the prior version of ISPF/PDF. It prevents a paragraph from being flowed beyond the screen width in the normal case, when default bounds are being used, and the TF command is entered without an ending column being specified.

You can restructure paragraphs containing lines which include DBCS strings based on the following rules:

- If a character in a DBCS string encroaches on the rightmost column position for the restructured text, the string is divided preceding that character. An SI character is added at the end of the line, and an SO character is added at the beginning of the new line.
- If BNDS boundaries are defined and a DBCS character is on the boundary, the DBCS character is involved in the text flow operation. An SO or SI character is added to both lines to ensure that DBCS character strings remain enclosed with SO and SI characters.
- If the MASK contains DBCS subfields, and some of the DBCS subfields cross the left, right, or both BNDS boundaries, the result is unpredictable.
- If a DBCS string crosses the left, right, or both BNDS boundaries, the result is unpredictable.
- When a text flow operation causes a subfield length of zero, the adjacent SO/SI or SI/SO character strings are removed.

If you are using the TF line command on a line in a formatted data set, considerations include:

- The current BNDS boundaries are automatically modified during command execution, and are reset to the original values after execution is complete. Modifications are as follows:
 - If the left boundary falls on the second byte of a DBCS character in a DBCS field, the boundary is shifted to the left by one byte.

- If the right boundary does not fall on the same field as the left boundary, it is set to point to the last byte of the field in which the left boundary falls. If it falls on the same DBCS field as the left boundary, and it also falls on the first byte of a DBCS character, the right boundary is shifted to the right by one byte.
- If you specify the column number with the text flow command, and if the column falls on the first byte of a DBCS character in a DBCS field, the column number increases by one.

A before-and-after example of text restructuring is shown in Figure 41. 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 --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- 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 41 (Part 1 of 2). Edit - Text Restructure Example

```
EDIT --- SPFDEVEL.PRM.SCRIPT(DEMO) - 01.01 ----- 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 41 (Part 2 of 2). Edit - Text Restructure Example

Uppercase (UC) Command

The Uppercase (UC) command translates a single line or multiple lines of text to uppercase. Enter the UC command in the line command area, typing over the line number. A number can follow the UC command to indicate that more than one line should be translated. For example, UC12 translates twelve lines to uppercase, starting with the line on which the command is entered. The format of the UC command is as follows:

```
UC      Translate one line to uppercase
UCn     Translate "n" lines to uppercase
UCC     Translate a block of lines to uppercase
```

You can translate a block of lines to uppercase by entering the line command UCC in the line command area of the first and last lines to be translated.

See Figure 42 on page 147 and Figure 43 on page 148 for examples of using the UC command.

Only alphabetic characters are affected by the UC command. Use of the UC command does not affect the caps mode of the data you are editing.

The following figure shows several examples of how the UC command is entered. Figure 43 on page 148 shows the results of these commands.

```

EDIT --- POEMPROJ.MAL.LMF(NURSERY) - 01.01 ----- COLUMNS 0001 0072
COMMAND ==>                                     SCROLL ==> HALF
***** ***** TOP OF DATA *****
000001 This line will not be translated.
UC0002 UC translates this line to uppercase.
000003 This line will not be translated.
UC2004 UC2 translates this line and the next line to uppercase.
000005 This line is translated by the UC2 on the previous line
000006 This line is not affected.
UCC007 UCC on this line and on line 13 translates lines
000008 7 through 13 to uppercase:
000009 including this line,
000010 this line,
000012 this line,
UCC013 through this line.
000014 The End.
000019
***** ***** BOTTOM OF DATA *****

```

Figure 42. Example of UC commands before processing

```

EDIT --- POEMPROJ.MAL.LMF(NURSERY) - 01.01 ----- COLUMNS 0001 0072
COMMAND ==>                                SCROLL ==> HALF
***** ***** TOP OF DATA *****
000001 This line will not be translated.
000002 UC TRANSLATES THIS LINE TO UPPERCASE.
000003 This line will not be translated.
000004 UC2 TRANSLATES THIS LINE AND THE NEXT LINE TO UPPERCASE.
000005 THIS LINE IS TRANSLATED BY THE UC2 ON THE PREVIOUS LINE
000006 This line is not affected.
000007 UCC ON THIS LINE AND ON LINE 13 TRANSLATES LINES
000008 7 THROUGH 13 TO UPPERCASE:
000009 INCLUDING THIS LINE,
000010 THIS LINE,
000012 THIS LINE,
000013 THROUGH THIS LINE.
000014 The End.
000019
***** ***** BOTTOM OF DATA *****

```

Figure 43. Results of processing the UC commands shown in the previous figure

Lowercase (LC) Command

Use the Lowercase (LC) command to translate one or more lines of text to lowercase. Enter the LC command in the line command area, typing over the line number. A number can follow the LC command to indicate that more than one line should be translated. For example, if you want to translate 5 lines to lowercase, you would type LC5 on the command line. This would translate 5 lines, starting with the line on which you typed the command. The format of the LC command is as follows:

```

LC      Translate one line to lowercase
LCn    Translate "n" lines to lowercase
LCC    Translate a block of lines to lowercase

```

To translate a block of lines to lowercase, enter the LCC command in the line command area of the first and last lines to be translated. The first and the second LCC command do not need to be on the same page; you can use scrolling between entering the first and the second LCC command.

Only alphabetic characters are affected by the LC command. Use of the LC command does not affect the caps mode of the data you are editing.

The following figure shows several examples of how the LC command is entered. Figure 45 on page 149 shows the results of these commands.

```

EDIT --- POEMPROJ.MAL.LMF(NURSERY) - 01.01 ----- COLUMNS 0001 0072
COMMAND ==>                                     SCROLL ==> HALF
***** ***** TOP OF DATA *****
LC0001 LC TRANSLATES ONLY THIS LINE TO LOWERCASE.
000002 THIS LINE IS NOT TRANSLATED TO LOWERCASE
000003 NOR IS THIS LINE TRANSLATED.
LC3004 LC3 TRANSLATES THIS LINE AND THE NEXT 2 LINES
000005 INCLUDING THIS LINE. NOTE: LC DOES NOT AFFECT NUMBERS OR
000006 SYMBOLS SUCH AS 1, 2, 3, &, % AND SO ON.
000007 THIS LINE IS NOT TRANSLATED TO LOWERCASE
000008 NOR IS THIS LINE
LCC009 LCC TRANSLATES ALL LINES UP THROUGH THE NEXT LINE WITH LCC
000010 IN THE COMMAND AREA
000012 THIS INCLUDES THIS LINE
000013 AND THIS LINE
LCC014 THROUGH THIS LINE
000019
***** ***** BOTTOM OF DATA *****

```

Figure 44. Example of LC commands before processing

```

EDIT --- POEMPROJ.MAL.LMF(NURSERY) - 01.01 ----- COLUMNS 0001 0072
COMMAND ==>                                     SCROLL ==> HALF
***** ***** TOP OF DATA *****
000001 lc translates only this line to lowercase.
000002 THIS LINE IS NOT TRANSLATED TO LOWERCASE
000003 NOR IS THIS LINE TRANSLATED.
000004 lc3 translates this line and the next 2 lines
000005 including this line. note: lc does not affect numbers or
000006 symbols such as 1, 2, 3, &, % and so on.
000007 THIS LINE IS NOT TRANSLATED TO LOWERCASE
000008 NOR IS THIS LINE
lcc009 lcc translates all lines up through the next line with lcc
000010 in the command area
000012 this includes this line
000013 and this line
lcc014 through this line
000019
***** ***** BOTTOM OF DATA *****

```

Figure 45. Results of processing the LC commands shown in the previous figure

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 overlaid. 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.

There is no special support for DBCS data handling. The user is responsible for DBCS data integrity when overlaying lines.

A before-and-after example of overlaying lines is shown in Figure 46. Generally, a list such as that shown in the figure would be maintained and edited as a single column (left adjusted). Prior to the overlay, portions of the list would be right-shifted by the appropriate amounts to overlay in the 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 MASK
000049 <, <N, <<, <<N (DATA LEFT)
000050 >, >N, >>, >>N (DATA RIGHT)
000051 (, (N, ((, ((N (COLUMNS LEFT)
000052 ), )N, )) , ))N (COLUMNS RIGHT)
000053 BOUNDS
000054
MM_
***** ***** BOTTOM OF DATA *****

```

Figure 46 (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 46 (Part 2 of 2). Edit - Overlay Line Example

Identifying Columns (COLS Command)

You can display a column 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 47. The characters '=COLS>' in the line command area are displayed by edit to identify the COLS line.

```

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 47. Edit - Column Identification 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.

Defining Masks (MASK Command)

You can use a mask with the I (insert), TE (text entry), and TS (text split) line commands while editing an unformatted data set. 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 the ENTER key, 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 48. 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.

Note: The MASK command is ignored in formatted edit mode. If you have defined a MASK prior to entering formatted edit mode, the mask is **not** retained in the current edit profile.

```

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 48 (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 48 (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 the ENTER key, 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 49 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 49. 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 the ENTER key, 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 set "type" (the lowest level qualifier in the data set name) and whether or not the data is sequence-numbered. The default left and right column boundaries are shown in Figure 50.

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

Figure 50. 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 51 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--
=BND>
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 51. 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.

Note: If you specify different shift amounts on the first and last lines of a block of text, and one of the shift amounts is the default, the lines will be shifted according to the non-default amount.

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 set "type" (lowest level qualifier of the data set name). 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.

If you are using the column shifting or data shifting line command while editing a formatted data set, special considerations include:

- The current BNDS boundaries are automatically modified during command execution, and are reset to the original values after execution is complete. Modifications are as follows:
 - If the left boundary falls on the second byte of a DBCS character in a DBCS field, the boundary is shifted to the left by one byte.
 - If the right boundary does not fall on the same field as the left boundary, it is set to point to the last byte of the field in which the left boundary falls. If it falls on the same DBCS field as the left boundary, and it also falls on the first byte of a DBCS character, the right boundary is shifted to the right by one byte.
- If you use the data shift or column shift line command to shift a DBCS field and you specify an odd-length shift amount, the shift amount will be decreased by one to preserve DBCS data integrity.

> and < (Data Shifting) Commands

Within the column boundaries, data shifting is accomplished by squeezing out multiple blank characters at one end of the data, and increasing the number of blanks at the other end. The following rules apply to data shifting:

- 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 with the following intensified warning message: **==ERR>**. You can remove the warning message with the RESET primary command, or you can overtype the message or data on that line.
- If a request to shift an odd number of bytes causes an odd-length DBCS string, the shift volume is decreased by one and the operation is performed. The line number is replaced with the following intensified warning message: **==ERR>**.
- If half of a DBCS character is involved in the shift, it is excluded from the operation.

Figure 52 shows a before-and-after example of data shifting. The block of lines, starting at line 1910 through line 1980, is shifted left four column positions.

```
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 52 (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 52 (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.

The following rules apply to column shifting when you shift lines that contain DBCS strings:

- If half of a DBCS character is involved in the shift, it is excluded from the operation; the shift count is changed automatically.
- If a column shift causes a DBCS string and an EBCDIC string to be connected, a shift-out or shift-in character, as appropriate, is inserted between the strings. The shift count is changed automatically.

- If left and/or right BNDS boundaries are set, a DBCS character may cross the boundary. The DBCS character that crosses the boundary is excluded from the operation, and the shift count is changed automatically.
- If a request to shift an odd number of columns causes an odd-length DBCS string, the requested shift number is discarded. The shift is executed up to the next subfield boundary within the BNDS boundary, if any. If no subfield boundary is found, the line number is replaced with the following intensified warning message: **==ERR>**, and the short message for an incomplete data shifting error is displayed.

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 the ENTER key, 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 53 shows a before-and-after example of excluded lines. The block of lines, from 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

```

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 53 (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 53 (Part 2 of 2). Edit - Excluded Lines Example

The S line command causes the block of excluded lines to be scanned, and one or more lines selected for redisplay. The selected lines are those with the 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.

Note: If the S line command results would cause only one line not to be displayed, then that line is displayed anyway. This could cause more lines to be displayed than you actually requested. For example, if there are only two lines excluded in the block, then an S1 command would cause both lines to be displayed.

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

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

Figure 54 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 as part of the data.

```

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 54 (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      BROLIST = 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      BROLIST = OFF;                    /*    TURN OFF LIST FLAG    */
003000      END;                              /*    END OF LOOP           */
003100      BROSMEMB = BLANKS;                 /*    CLEAR SELECTED MEMBER */
003200      END;                              /*                          */
***** ***** BOTTOM OF DATA *****

```

Figure 54 (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, ISRSLIB. 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 data set 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 LABEL command has been issued. You can determine whether you want the notes displayed during the edit session by using the NOTE command or the NOTE or NONOTE operand on the MODEL command.

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 55. 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 data set 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.

----- 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
9	PASCAL	- ISPF services in PASCAL programs

Enter END command to cancel MODEL command.

Figure 55. Edit - Model Class Selection Panel

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 is the PDF library type or the last qualifier in the data set name. For example, you normally store panels in data sets 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 data set with a type qualifier of PANELS. Assume, however, that you call your panels "screens" and maintain them in a data set 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 data set being edited by the MODEL command. The MODEL command allows you to specify either:

- The class of models to be searched when retrieving models
- The logical name of the model to be copied and, optionally, the destination of the model.

The first form of the command is used to specify the class of models to be used. This form has the following format:

```
MODEL [CLASS [class-name]]
MOD
```

CLASS classname 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 or an allowable abbreviation. 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.

The second form of the MODEL command is used to copy models into the current file. This form has the following format:

```
MODEL [model-name [qualifier...]] [AFTER label] [NOTE ]
MOD [BEFORE label] [NONOTE]
```

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. Do not create a model with the model-name of "class" since this is a reserved word on the MODEL command.

AFTER label	identifies the line after which the model is to be copied.
BEFORE label	identifies the line before which the model is to be copied.
NOTE	overrides the current profile note mode, to include any notes which are part of the model.
NONOTE	overrides the current profile note mode, to exclude any notes which are part of the model.

If a label is not entered, a line command "A" or "B" must be used to specify the destination of the model.

Using Models

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

To use a model, first edit your data set. Then, determine where you wish to place the model. If you are editing an existing data set, define a label or use the A (After) or B (Before) line command to indicate where the model will go. You need not use the label or the A or B command when you have a new data set. 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.

As an example, assume that you are writing a dialog function using CLIST commands, and you want to have the CLIST display a panel. You are editing your CLIST member, called ISPFPROJ.FLG.CLIST(DEMO1). Since your data set type (CLIST) 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 56).

----- COMMAND MODELS -----

OPTION ==> _

VARIABLES	TABLES (General)	TABLES (Row)	LIBRARY ACCESS
V1 VGET	G1 TBCREATE	R1 TBADD	L1 LMCLOSE
V2 VPUT	G2 TBOPEN	R2 TBDELETE	L2 LMERASE
	G3 TBQUERY	R3 TBGET	L3 LMFREE
	G4 TBSAVE	R4 TBPOT	L4 LMGET
DISPLAY	G5 TBCLOSE	R5 TBMOD	L5 LMINIT
D1 DISPLAY	G6 TBEND	R6 TBEXIST	L6 LMMADD
D2 TBDISPL	G7 TBERASE	R7 TBSARG	L7 LMMDEL
D3 SETMSG	G8 TBSTATS	R8 TBSCAN	L8 LMMFIND
D4 PQUERY		R9 TBTOP	L9 LMMLIST
	MISCELLANEOUS	R10 TBBOTTOM	L10 LMMREN
	M1 SELECT	R11 TBSKIP	L11 LMMREP
FILE TAILORING	M2 CONTROL	R12 TBVCLEAR	L12 LMOPEN
F1 FTOPEN	M3 BROWSE	R13 TBSORT	L13 LMPROM
F2 FTINCL	M4 EDIT		L14 LMPUT
F3 FTCLOSE	M5 LOG		L15 LMQUERY
F4 FTERASE	M6 GETMSG		L16 LMRENAME

Enter END command to cancel MODEL command.

Figure 56. Edit - Model Selection Panel for CLIST Models

If you then select option D1 (DISPLAY), the model for the display service (Figure 57) is inserted in your CLIST at the location you've indicated with a label or 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 CLIST 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 SPECIAL 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 57. 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 data set 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 58.

```

      I I I I I I I I I I
        I I
          I I
            I I
              I I
                I I
      I I I I I I I I I I
)N
)N  the letter I for logo

```

Figure 58. 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 59.

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

PANEL FORMATS:

- | | | | | | |
|----|-----------|-------------------|----|----------|-----------------|
| F1 | ENTRY | - Data entry | F4 | TBDISPL | - Table Display |
| F2 | MULTIPLE | - Multiple column | F5 | TUTORIAL | - Help/tutorial |
| F3 | SELECTION | - Menu selection | | | |

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 |
| SA | REFRESH | - Refresh variables prior to redisplay |
| SB | ATTRIBA | - New attribute character definition for areas |

Enter END command to cancel MODEL command.

Figure 59. Edit - Panel Model Selection Panel

Copy this panel into your panel data set and modify it, as shown in Figure 60, by adding a new format, F6. Note the addition of Format F6, BLOCKLTR.

```

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

PANEL FORMATS:
F1 ENTRY      - Data entry           F4 TDISPL     - Table Display
F2 MULTIPLE   - Multiple column      F5 TUTORIAL   - Help/tutorial
F3 SELECTION  - Menu selection        F6 BLOCKLTR   - Block letter format

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
SA REFRESH    - Refresh variables prior to redisplay
SB ATTRIBA    - New attribute character definition for areas

```

Enter END command to cancel MODEL command.

Figure 60. Edit - Modified Visible Panel Model Selection Panel

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 61.

This concept could be used to allow each user to have a set of individual models, and allow the installation to have its own set of general models, without having multiple copies of the PDF model selection panels. For each model class, the installation could provide two additional entries on the selection panel: one for installation-wide models; and one for user models. Each entry could point to a selection panel, with each user having a copy of the user selection panel to customize for individual use.

```

)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(ISREMSSEL)'
    MULTIPLE, 'PANEL(ISREMSSEL)'
    F3, 'PGM(ISRECMBR) PARM(ISREMMF3)'
    SELECT, 'PGM(ISRECMBR) PARM(ISREMMF3)'
    SELECTIO, 'PGM(ISRECMBR) PARM(ISREMMF3)'
    F4, 'PGM(ISRECMBR) PARM(ISREMMF4)'
    TBDISPL, 'PGM(ISRECMBR) PARM(ISREMMF4)'
    F5, 'PGM(ISRECMBR) PARM(ISREMMF5)'
    TUTORIAL, 'PGM(ISRECMBR) PARM(ISREMMF5)'
    F6, 'PANEL(BLKLTRS)'
    BLOCKLTR, 'PANEL(BLKLTRS)'
    S1, 'PANEL(ISREMASG)'
    ASSIGN, 'PANEL(ISREMASG)'
    etc. etc. etc.

```

/* NEED TO QUALIFY THIS */

/* PANEL FOR COLUMNS ID. */

/* AUTOMATIC SINGLE COLUMN*/

/* FOR 8 OR LESS SELECTION*/

/* LARGE BLOCK LETTERS */

/* ASSIGN AND VERIFY */

/* NEED QUALIFIERS */

Figure 61. 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 62. Note particularly the)INIT and)PROC sections of the coding.

```

)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 (&ISRMSPL = '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
  &ISRMEND = '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 (&ISRMSPL = 'RETURN ') /* MAKE SURE ITS NOT END ON MBR. */
        &ISRMEND = 'Y' /* NO - ITS BECAUSE USER HIT END */
)END

```

Figure 62. Edit - Block Letter Model Selection Panel 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. For example, in 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 62 for all new panels.

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 F. 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 61, 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 data set. 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 57).

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 utilities option provides a variety of functions for library, data set, and catalog maintenance: moving and copying data; resetting library statistics; initiating hardcopy output; displaying or printing VTOC entries for a DASD volume; browsing and printing held SYSOUT data; creating and updating an application command table; converting old format panels and messages; and defining and maintaining formats for browsing and editing formatted data sets.

The utility selection panel is shown in Figure 63.

```

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

 1 LIBRARY      - Library utility:
                  Print index listing or entire data set
                  Print, rename, delete, or browse members
                  Compress data set
 2 DATASET      - Data set utility:
                  Display data set information
                  Allocate, rename, or delete entire data set
                  Catalog or uncatalog data set
 3 MOVE/COPY    - Move, copy, or promote members or data sets
 4 DSLIST       - Data set list:
                  Print or display (to process) list of data set names
                  Print or display VTOC information
 5 RESET        - Reset statistics for members of ISPF library
 6 HARDCOPY     - Initiate hardcopy output
 7 VTOC         - Display or print VTOC entries for a DASD volume
 8 OUTLIST      - Display, delete, or print held job output
 9 COMMANDS     - Create/change an application command table
10 CONVERT      - Convert old format messages/menus to new format
11 FORMAT       - Format definition for formatted data edit/browse

```

Figure 63. Utility Selection Panel

The following describes each of the utility functions.

LIBRARY UTILITY (OPTION 3.1)

When this option is selected, a panel is displayed that allows you to specify a data set and indicate an action to be performed. Figure 64 shows a panel on which data set name SPFDemo.XXX.COBOL has been entered.

For information about specifying the library or data set name, see Chapter 3, "Libraries and Data Sets" on page 25. You can specify concatenation of up to four groups on the library utility panel.

Possible actions are:

blank	- Display member list
C	- Compress data set
X	- Print index listing
L	- Print entire data set
I	- Display data set information
B	- Browse member
P	- Print member
R	- Rename member
D	- Delete member
S	- Display data set information (short)

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

blank - Display member list          B - Browse member
C - Compress data set                P - Print member
X - Print index listing               R - Rename member
L - Print entire data set            D - Delete member
I - Data set information              S - Data set information (short)

ISPF LIBRARY:
PROJECT ==> SPFDemo
GROUP   ==> XXX           ==>           ==>
TYPE    ==> COBOL
MEMBER  ==>                (If option "P", "R", "D", or "B" selected)
NEWNAME ==>                (If option "R" selected)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>          (If not cataloged)

DATA SET PASSWORD ==>      (If password protected)
```

Figure 64. Library Utility Panel

Library Utility (Option 3.1)

This utility is intended primarily for maintenance of partitioned data sets, but the print index listing function (option X) and the print entire data set function (option L) also apply to sequential data sets.

If you leave the option field blank (to display a member list), you must specify a partitioned data set. 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 scrolling commands or the LOCATE command. You terminate the member list by entering the END command.

If you select option C (compress data set), any partitioned data set may be specified. The data set is compressed by invoking the IEBCOPY utility.

Note: Use of the compress option requires the TSO/E Release 2 Program Product or a special installation procedure, described in ISPF and ISPF/PDF for MVS Installation and Customization.

If you select option X (print index listing), a DASD-resident sequential or partitioned data set may be specified. The index listing is recorded in the ISPF list file. For a partitioned data set, the index listing includes general information about the data set followed by a member list. For a sequential data set, the index listing includes general information only. See Appendix B, "PDF Listing Formats" on page 323 for a discussion of the index listing format.

Note: If PDF was entered in TEST mode, the listing also includes TTR data for each member of the data set.

If you select option L (print entire data set), a DASD-resident sequential or partitioned data set may be specified. The allowable data set characteristics are the same as for browse, except that the data is truncated at 300 characters if the LRECL exceeds this length. An index listing and source listing of the entire data set (including all members of a partitioned data set) are recorded in the ISPF list file.

If you select option P (print member), a partitioned data set and member name must be specified. The allowable data set characteristics are the same as for the L option. A source listing of the member is recorded in the ISPF list file.

If you select option R (rename member) or option D (delete member), a partitioned data set and member name must be specified. You must also specify a new member name for the "rename member" function.

If you select option B (browse member), a partitioned data set and member name must 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 select option I or S (display data set information), information about the selected data set is displayed. The information displayed by option I is the same information displayed by the data set utility (option 3.2). The information displayed by option S is the same except that it lacks the size of the directory and the number of members for partitioned data sets. For sequential data sets, options I and S display the same information.

Figure 65 shows a before-and-after example where members ACCT1 and ACCT2 are printed, member UPDATE is deleted, and member LISTNEW is renamed LISTOUT. This examples shows the date the member was created and the date the member was last modified. The date is in the national format (yy/mm/dd). You can define this format in the national date variable.

Note: If any members are to be printed, the data set characteristics must conform to those for the L option.

```

LIBRARY  SPFDEMO.MYLIB.PLI -----
COMMAND ==>
      NAME      RENAME    VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD   ID
ACCOUNT
P ACCT1          01.01   83/02/11  83/04/23 14:52    199   193   0    BECKETT
P ACCT2          01.00   83/03/09  83/03/09 17:07     20    20    0    BECKETT
COINS           01.04   83/04/24  83/04/28 16:20     19    19    4    JOSLIN
COMPX           01.00   83/01/21  83/01/21 11:08     44    44    0    JOSLIN
COMPY           01.01   83/01/14  83/01/16 12:30     13    13    1    JOSLIN
DCLS            01.00   83/04/23  83/04/23 15:14     20    20    0    LEESB
R LISTNEW  LISTOUT  01.02   83/04/23  83/05/06 10:00     17    13    6    LEESB
MAIN            01.00   83/01/09  83/01/09 17:08      4     4     0    KIRK
TESTDIR         01.02   83/04/23  83/05/06 09:04     30    43    10   LAIDLAW
D UPDATE        01.00   81/03/26  81/04/01 13:08     13    13    0    COONS
**END**

```

Figure 65 (Part 1 of 2). Library Utility - Print, Rename and Delete

```
LIBRARY  SPFDEMO.MYLIB.PLI -----
COMMAND ==>
  NAME      RENAME  VV.MM  CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT                                01.00  83/01/09  83/01/09 17:07    21   21   0  HUNTER
ACCT1      *PRINTED 01.01  83/02/11  83/04/23 14:52   199  193   0  BECKETT
ACCT2      *PRINTED 01.00  83/03/09  83/03/09 17:07    20   20   0  BECKETT
COINS                                01.04  83/04/24  83/04/28 16:20    19   19   4  JOSLIN
COMPX                                01.00  83/01/21  83/01/21 11:08    44   44   0  JOSLIN
COMPY                                01.01  83/01/14  83/01/16 12:30    13   13   1  JOSLIN
DCLS                                01.00  83/04/23  83/04/23 15:14    20   20   0  LEESB
LISTNEW    *RENAMED
MAIN                                01.00  83/01/09  83/01/09 17:08     4    4   0  KIRK
TESTDIR    01.02  83/04/23  83/05/06 09:04    30   43  10  LAIDLAW
UPDATE     *DELETED
***END***
```

Figure 65 (Part 2 of 2). Library Utility - Print, Rename and Delete

DATA SET UTILITY (OPTION 3.2)

When you select this option, a panel is displayed that allows you to specify a data set and indicate an action to be performed (Figure 66). Possible actions are:

- A - Allocate new data set
- R - Rename data set
- D - Delete data set
- C - Catalog data set
- U - Uncatalog data set
- blank - Display data set information
- S - Display data set information (short)

You may specify any DASD-resident sequential or partitioned data set for option A (allocate new data set). You may specify any DASD-resident, non-VSAM data set for the other options. For information about specifying library and data set names, see Chapter 3, "Libraries and Data Sets" on page 25.

```
----- DATA SET UTILITY -----
OPTION ==> _

A - Allocate new data set          C - Catalog data set
R - Rename entire data set        U - Uncatalog data set
D - Delete entire data set        S - Data set information (short)
blank - Data set information

ISPF LIBRARY:
PROJECT ==> SPFDEMO
GROUP   ==> MYLIB
TYPE    ==> PLI

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>          (If not cataloged, required for option "C")

DATA SET PASSWORD ==>      (If password protected)
```

Figure 66. Data Set Utility Panel

Data Set Utility (Option 3.2)

If you select option A (allocate new data set), a panel is displayed to allow you to specify a volume serial (you may specify a real DASD volume or an IBM 3850 virtual volume), a generic unit type (like SYSDA) for allocation to a default volume, data set characteristics, and space allocation parameters (Figure 67). Default values are pre-entered (except for expiration date) based on what you last entered on this panel, or based on the last display data set information request (whichever occurred most recently). You may overtype the displayed defaults. When you press ENTER, the new data set is allocated and cataloged. Entering the END command returns you to the previous panel without performing the allocation.

Note: To allocate a data set to a 3850 virtual volume, you must also have MOUNT authority (see the TSO ACCOUNT command).

For partitioned data sets, you must specify the number of directory blocks. Each 256-byte block accommodates the following number of directory entries:

- Data sets with ISPF statistics: 6
- Data sets without ISPF statistics: 21
- Load module data sets: 4 to 7 (depending upon attributes)

```
----- ALLOCATE NEW DATA SET -----  
COMMAND ==>  
  
DATA SET NAME: NEWPROJ.MASTER.ASM  
  
VOLUME SERIAL   ==> -           (Blank for authorized default volume)  
GENERIC UNIT    ==> -           (Generic group name or unit address)  
SPACE UNITS     ==> TRKS       (BLKS, TRKS or CYLS)  
PRIMARY QUAN    ==> 50        (in above units)  
SECONDARY QUAN  ==> 0         (in above units)  
DIRECTORY BLOCKS ==> 20       (Zero for sequential data set)  
RECORD FORMAT   ==> FB  
RECORD LENGTH   ==> 80  
BLOCK SIZE      ==> 1600  
EXPIRATION DATE ==> 83/05/24   (national format or blank)
```

Figure 67. Data Set Utility - Allocate New Data Set

If you select option R (rename data set), a panel is displayed to allow you to enter the new data set name (Figure 68). Type the new data set name and press ENTER to rename the data set, or enter the END command to cancel it. Either action causes a return to the previous panel.

If you specify a volume serial for a data set to be renamed, PDF checks to see whether the data set is cataloged on that volume. If it is, the rename panel prompts you to specify whether to recatalog the data set. The displayed default for that field is YES. If you specify a volume serial and the data set is not cataloged, the new name will not be cataloged. If a volume serial is not specified, the data set is recataloged to the new data set name and the old data set name is uncataloged.

If you are renaming a GDG (Generation Data Group) data set which has been defined with scratch and limit parameters, where the limit has been reached, you must specify a volume serial. The new data set name can then be cataloged with option C.

If you select option D (delete data set), a confirmation panel is displayed to ensure that you did not select this option by mistake. As directed in the panel, press ENTER to confirm deletion or enter the END command to cancel your request. Either action causes a return to the previous panel.

```
----- RENAME DATASET -----  
COMMAND ==>  
  
DATA SET NAME: SPFDEMO.XXX.COBOL  
VOLUME:          010901  
  
ENTER NEW NAME BELOW:      (The new name will also be recataloged.)  
  
ISPF LIBRARY:  
  PROJECT ==>  
  GROUP   ==>  
  TYPE    ==>  
  
OTHER PARTITIONED OR SEQUENTIAL DATA SET:  
  DATA SET NAME ==> 'NEWPROJ.DEVEL.COBOL' _
```

Figure 68. Data Set Utility - Rename Data Set

If you specify a volume serial for the data set to be deleted, PDF checks to see whether the data set is cataloged on that volume. If so, the confirm delete panel prompts you to specify whether to uncatalog the data set. The displayed default is YES. If no volume serial is specified, the data set is scratched and uncataloged.

Note: PDF will not allow deletion of a password protected data set.

If you select option C (catalog data set), the specified data set is cataloged. For this option you must specify the volume serial on which the data set resides (regardless of whether the data set is specified via project, library, and type or "other" data set name). The data set must reside on the specified volume.

If you select option U (uncatalog data set), the specified data set name is uncataloged. There is no requirement for the specified data set to be allocated or mounted.

If you leave the option field blank (to display data set information), the following information is displayed (Figure 69):

- Volume serial
- Device type
- Data set organization
- Record format
- Record length
- Block size
- Creation date
- Expiration date
- First extent size
- Secondary allocation quantity
- Space (allocated and used)
- Number of extents (allocated and used)
- Directory blocks (if PDS, allocated and used)
- Number of members (if PDS)

To return to the previous panel, press the ENTER key or enter the END command.

If you select option S (to display short data set information), the same information is displayed as for the long display information option (Figure 69) except that it does not include the size of the directory or the number of members for partitioned data sets. For sequential data sets, the long and short forms are identical. To return to the previous panel, press the ENTER key or enter the END command.

```
----- DATA SET INFORMATION -----  
COMMAND ==> _  
  
DATA SET NAME: SPFDEMO.MYLIB.PLI  
  
GENERAL DATA:                                CURRENT ALLOCATION:  
  Volume serial:          D00163              Allocated tracks:      20  
  Device type:           3330                Allocated extents:     1  
  Organization:          PO                  Maximum dir. blocks:   10  
  Record format:         FB  
  Record length:         80  
  Block size:            1600  
  1st extent tracks:     20  
  Secondary tracks:      0  
  
  Creation date:         82/10/14  
  Expiration date:      83/05/25  
  
                                CURRENT UTILIZATION:  
                                Used tracks:           5  
                                Used extents:           1  
                                Used dir. blocks:       2  
                                Number of members:     10
```

Figure 69. Data Set 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" data set (and member if it is partitioned), and to indicate an action to be performed (Figure 70). Possible actions are:

- C - Copy data set or member(s)
- M - Move data set or member(s)
- P - Promote data set or member(s)
- CP - Copy and print
- MP - Move and print
- PP - Promote and print

```
----- MOVE/COPY UTILITY -----
OPTION ==> C_

C - Copy data set or member(s)          CP - Copy and print
M - Move data set or member(s)         MP - Move and print
P - Promote data set or member(s)      PP - Promote and print

SPECIFY "FROM" DATA SET BELOW, THEN PRESS ENTER KEY

FROM ISPF LIBRARY:
PROJECT ==> SPFDEMO
GROUP   ==> MYLIB
TYPE    ==> PLI
MEMBER  ==>                (Blank for member list, * for all members)

FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>          (If not cataloged)

DATA SET PASSWORD ==>      (If password protected)
```

Figure 70 (Part 1 of 2). Move/Copy Utility Panels

```
COPY --- FROM SPFDEMO.MYLIB.PLI -----
COMMAND ==>

SPECIFY "TO" DATA SET BELOW

TO ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  GROUP   ==> MASTER_
  TYPE    ==> PLI

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>
  VOLUME SERIAL ==>           (If not cataloged)

DATA SET PASSWORD ==>           (If password protected)

"TO" DATA SET OPTIONS:
  IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS ==> YES (YES or NO)
  IF SEQUENTIAL, "TO" DATA SET DISPOSITION ==> OLD (OLD or MOD)
  SPECIFY PACK OPTION FOR "TO" DATA SET ==> NO (YES, NO, or BLANK)
```

Figure 70 (Part 2 of 2). Move/Copy Utility Panels

When you press ENTER, a second panel is displayed, depending on whether you are moving or copying, or promoting. If you specify move or copy, the second panel allows you to specify the "to" data set (Part 2 of Figure 70). You must also specify the following options on this panel:

- Replace like-named members (YES or NO), if the "to" data set is partitioned.
- "To" data set disposition (OLD or MOD), if the "to" data set is sequential.
- Packing option for the "to" data set (YES or NO), to specify whether to store the moved or copied data in packed format. If you leave the field blank, the data is stored as it exists in the "from" data set.

The compression technique ISPF uses to pack data is unique to ISPF. Attempts to access or execute the data outside of PDF can cause unpredictable results. See "Packing Data (PACK Command)" on page 88 for more information.

Note: If you have selected several members of an ISPF library or a partitioned data set to be moved or copied to a sequential data set, the members are written to the "to" data set one after another. The "to" data set disposition controls only the initial positioning of the "to" data set.

If you specify promote, the second panel allows you to specify the "to" data set (Figure 71). This second panel appears only when you are promoting from a private library into an entry-level controlled library or when you are freeing a previously locked member of a controlled library. You must also specify the following options on this panel:

ACTION This field specifies whether a normal promotion of the member is to occur, or the member is to be made available for update (freed). The latter action may be specified if you do not want to promote the member for some reason, but want to make the member available to others. The specified actions are:

MOVE Promotion in which the member is moved to the target library and deleted from the private library. The member becomes available for update (no longer locked by your logon id).

FREE Make the member available, no longer locked by your logon id. The member is not moved to the target library, but is deleted from the private library.

REASON CODE This is a 26-character field that you can use as you wish. This field will be in the activity log. You can use it for tracking, testing in a user exit, or for any other purpose.

PACK DATA You can use this field to override the library control. Leaving this field blank defaults to the packed data library control specification. If the data is packed (or unpacked) when promoted into an entry-level library, it will remain packed (or unpacked) throughout each subsequent promotion.

For additional information about the promotion process, see ISPF/PDF for MVS Library Management.

For move or copy, you may specify a DASD-resident sequential or partitioned data set for either the "from" or "to" data set. For promote, the "from" data set may be sequential, but the "to" data set must be a controlled ISPF library. Use the Library Management Utilities (option 8) to set up controls for libraries. The allowable data set characteristics are the same as for browse. You can move, copy, or promote load modules (stored in partitioned data sets with "undefined" record format), but you cannot print them.

```
PROMOTE --- FROM SPFDEMO.MYLIB.PLI (ABCD) -----  
COMMAND ==>  
  
SPECIFY "TARGET" CONTROLLED LIBRARY BELOW.  
  
TO ISPF LIBRARY:  
  PROJECT ==> SPFDEMO  
  GROUP   ==> MASTER  
  TYPE    ==> PLI  
  MEMBER  ==>          (Blank unless member is to be renamed)  
  
ACTION    ==> MOVE      (MOVE to copy and delete source  
                        FREE to only delete source)  
  
REASON CODE ==>  
  
PACK DATA ==>          (YES, NO, or blank to default to library controls)
```

Figure 71. Promotion "To" Panel

Both data sets must already exist. The utility does not automatically allocate space for a new "to" data set.

If you select M (move) and the "from" data set is partitioned, or if you select P (promote), the selected member(s) are deleted from the "from" data set after they have been successfully copied to the "to" data set. If you indicate M (move) and the "from" data set is sequential, the entire "from" data set is deleted after its contents have been successfully copied to the "to" data set. If you select C (copy), no deletions occur.

If you select either MP (move and print), CP (copy and print), or PP (promote and print), source listing(s) are recorded in the ISPF list file. If the "to" data set is partitioned, a listing of each new or replaced member is recorded. For move or copy, if the "to" data set is sequential, a listing of its entire contents is recorded after the information has been successfully copied. The allowable data set characteristics for printing are the same as for browse.

The following rules apply for specifying member names, if the "from" data set is partitioned:

- Specifying a valid "from" member name causes a single member to be moved, copied, or promoted.
- 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, copied, or promoted.

If the "to" data set is partitioned, you must specify the "to" member name for the following conditions:

- The "from" data set is sequential.
- A single member is to be moved, copied, or promoted from a partitioned data set and the member is to be renamed in the "to" data set. (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" data set is partitioned and the "from" member name is left blank. You may select members to be moved, copied, or promoted by entering an S in front of one or more member names. The options for move, copy, or promote and print or no print must have been previously specified on the first move/copy panel. If the "to" data set is partitioned, you can also rename the member(s) by entering a new member name in the field immediately following the current name. For promote, you must have authority to promote the member under the new name. A member with the old name in the "to" library is not replaced or freed.

You can 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 72 shows a before-and-after example in which members ACCOUNT, ACCT1, ACCT2, and COMPY are copied to another partitioned data set, and member COMPY is also renamed ZCOMP.

```
COPY --- FROM SPFDEMO.MYLIB.PLI TO SPFDEMO.MASTER.PLI -----
COMMAND ==>                                SCROLL ==> PAGE
  NAME      RENAME      VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
S ACCOUNT                                01.00  83/01/09  83/01/09 17:07    21   21   0  HUNTER
S ACCT1                                01.01  83/02/11  83/04/23 14:52   199  193   0  BECKETT
S ACCT2                                01.00  83/03/09  83/03/09 17:07    20   20   0  BECKETT
  COINS                                01.04  83/04/24  83/04/28 16:20    19   19   4  JOSLIN
  COMPX                                01.00  83/01/21  83/01/21 11:08    44   44   0  JOSLIN
S COMPY      ZCOMP_          01.01  83/01/14  83/01/16 12:30    13   13   1  JOSLIN
  DCLS                                01.00  83/04/23  83/04/23 15:14    20   20   0  LEESB
  LISTOUT                               01.02  83/04/23  83/05/06 15:00    17   13   6  LEESB
  MAIN                                01.00  83/01/09  83/01/09 17:08     4    4   0  KIRK
  TESTDIR                               01.02  83/04/23  83/05/06 09:04    30   43  10  LAIDLAW
**END**
```

Figure 72 (Part 1 of 2). Move/Copy Utility - Copy Members from List


```

COPY --- FROM SPFDEMO.MYLIB.PLI TO SPFDEMO.MASTER.PLI -----
COMMAND ==>
      NAME      RENAME  VV.MM   CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT  *COPIED  01.00  83/01/09  83/01/09 17:07    21    21    0  HUNTER
ACCT1    *COPIED  01.01  83/02/11  83/04/23 14:52   199   193    0  BECKETT
ACCT2    *COPIED  01.00  83/03/09  83/03/09 17:07    20    20    0  BECKETT
COINS                    01.04  83/04/24  83/04/28 16:20    19    19    4  JOSLIN
COMPX                    01.00  83/01/21  83/01/21 11:08    44    44    0  JOSLIN
COMPY    *COPIED  01.01  83/01/14  83/01/16 12:30    13    13    1  JOSLIN
DCLS                    01.00  83/04/23  83/04/23 15:14    20    20    0  LEESB
LISTOUT                    01.02  83/04/23  83/05/06 15:00    17    13    6  LEESB
MAIN                    01.00  83/01/09  83/01/09 17:08     4     4    0  KIRK
TESTDIR                    01.02  83/04/23  83/05/06 09:04    30    43   10  LAIDLAW
***END***

```

Figure 72 (Part 2 of 2). Move/Copy Utility - Copy Members from List

For any move or copy operation that does not involve load modules, the following rules apply:

- If the "to" data set 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 partitioned data set may be moved or copied to a sequential data set by using a member list or specifying all members (*). The merged members appear in the "to" data set, replacing the original contents of the data set (if disposition OLD was specified) or added to the end (if disposition MOD was specified).
- If the "to" data set is partitioned, 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).

Move/Copy Utility (Option 3.3)

For any move, copy, or promote operation that does not involve load modules, the following rules apply:

- For move or copy, the record formats, block sizes, and logical record lengths for the two data sets need not be alike. When necessary, data is truncated or right-padded with blanks.
- For promote, if the record format is fixed, the LRECL must be the same, but the blocksize can be different. For promote with a variable record format, the target LRECL must be larger than the source LRECL. The blocks can differ but must be logically consistent with the LRECL.
- The data to be moved, copied, or promoted is not renumbered or modified in any way, except for possible truncation or padding as noted above. Printer control characters, if present, are treated as part of the data.
- Library statistics or other user data in directory entries are not modified when moving or copying between partitioned data sets.
- Partitioned data set members with user TTRs may not be moved, copied, or promoted.

For a move, copy, or promote of load modules, the following rules apply:

- Both data sets must be partitioned and must have undefined record format (RECFM=U). New members are added to the "to" data set except, for move or copy, when the member names are alike; like-named members are replaced (if replace YES was specified) or not copied (if replace NO was specified).
- No reblocking occurs. The block size of the "to" data set must be equal to or larger than the block size of the "from" data set. The logical record length (LRECL) is ignored.
- Load modules which were created for "planned overlay" may not be moved or copied.
- The print option, if specified, is ignored.

Alias entries may be moved or copied from one partitioned data set to another under the following conditions:

- When all members are specified (member name "*") and the move or copy is completed successfully for the main member(s) to which the aliases refer.
- From a member list:
 - When the main member(s) are selected first, are not renamed, and are successfully moved or copied; and
 - The alias name(s) are then selected without leaving the list.

In all other cases for move and copy, and in all cases for promote, you may select alias names but they are not preserved as aliases in the "to" data set (that is, the member(s) to which they refer are moved or copied and the alias entries are stored in the "to" data set with the alias flags turned off).

DATA SET LIST UTILITY (OPTION 3.4)

When you select this option, a panel (Figure 73) is displayed to allow you to specify a list of ISPF libraries or TSO data sets to be displayed in scrollable format or to be printed. The panel also allows you to display or print VTOC information. The allowable selections are:

blank Display data set list
V Display VTOC information
P Print data set list
PV Print VTOC information

You may specify any DASD-resident, non-VSAM data set.

```

----- DATA SET LIST UTILITY -----
OPTION ==>

Blank - Display data set list *      P - Print data set list
V    - Display VTOC information only  PV - Print VTOC information only

Enter one or both of the parameters below:
DSNAME LEVEL ==> z77crc
VOLUME       ==>

SPECIFY THE FOLLOWING, IF DISPLAYING A LIST OF DATA SETS:
DISPLAY FORMAT OPTION ==> SHORT  (QUICK, SHORT or LONG)
CONFIRM DELETE REQUEST ==> YES   (YES or NO)

* The following options will be available when the list is displayed

B - Browse data set          C - Catalog data set
E - Edit data set            U - Uncatalog data set
D - Delete entire data set   P - Print entire data set
R - Rename entire data set   X - Print index listing
I - Data set information     M - Display member list
S - Data set information (short) Z - Compress data set

```

Figure 73. Data Set List Utility Panel

If you select option P (print data set list) or leave the option field blank (display data set list), you must specify a DSNAME LEVEL with or without a VOLUME. If you specify a VOLUME, the list is created by searching the VTOC; if you do not specify a VOLUME, the list is created by searching the catalog. You must also specify the list format (QUICK,

SHORT, or LONG), and whether a confirmation panel is to be displayed if the delete option is entered from the list.

When you specify DSNAME LEVEL, you are defining the level qualifiers for the data set names to be included in the list. You can specify these qualifiers fully or partially, or default them. You must specify the first-level qualifier fully. Do not enclose the DSNAME LEVEL value in quotes. If you use an asterisk (*) to partially specify a qualifier, you must specify it as the first or last character in the qualifier. For example:

```
ISPFDEVL.*.SOURCE
SYS1.*LIB
```

In the first example, all data sets with ISPFDEVL as the first qualifier and SOURCE as the third qualifier will be included in the list. There is no restriction on the second qualifier or any qualifiers beyond the third one. In the second example, all data sets with a first qualifier of SYS1 and with a second qualifier ending in LIB will be included.

Figure 74 shows a typical data set list display (SHORT format). The list is sorted by data set name.

```
DATA SETS BEGINNING WITH Z77CRC -----
COMMAND ==>                                SCROLL ==> PAGE
```

DATA SET NAME	VOLUME	ORG	TRKS	%USED	XT
Z77CRC.AMCBS.BASE	MIGRAT				
Z77CRC.AMCBS.ICF	TSOPK1	PS	1	100	1
Z77CRC.CBLK	TSOPK2	PS	4	100	1
Z77CRC.CLIST	GB5311	PO	2	20	1
Z77CRC.DATASET.UTIL	GB5311	PS	4	100	1
Z77CRC.DAO	MIGRAT				
Z77CRC.EMPTY	GB5311	PO	1	0	1
Z77CRC.INDEX	TSOPK3	PS	77	100	1
Z77CRC.ITSPFLPG.OPTIONS	MIGRAT				
Z77CRC.ITSPFLPG.SCRIPT	MIGRAT				
Z77CRC.JASR2.PANELS	MIGRAT				
Z77CRC.LOC	MIGRAT				
Z77CRC.LOG.MISC	MIGRAT				
Z77CRC.NAMES.TEXT	GB5311	PS	1	100	1
Z77CRC.NET.MSG	GB5311	PS	1	100	1
Z77CRC.OPT34	MIGRAT				
Z77CRC.PANELS	MIGRAT				
Z77CRC.PDF.DEPEND	GB5311	PS	12	100	3
Z77CRC.RENAME	MIGRAT				

Figure 74. Data Set List Display (Short Format)

Data Set List Utility (Option 3.4)

The information shown for each data set is:

DATA SET NAME Data set name (as in the VTOC)

VOLUME Volume serial number

ORG Data set organization, one of the following:

- PS** Sequential (PSU is unmovable)
- PO** Partitioned (POU is unmovable)
- DA** Direct (DAU is unmovable)
- IS** Indexed sequential (ISU is unmovable)
- VS** VSAM
- blank None of the above data set organizations

TRKS Number of tracks allocated to the data set

%USED Percent of allocated tracks used (whole numbers, not rounded, except that if any track is used the minimum percentage is 1)

XT Number of extents allocated to the data set

The LONG format list display has two lines per data set. It includes the same information as the short form, plus the following additional information:

RECFM Record format

LRECL Logical record length

BLK SIZE Blocksize

CREATED Creation date in the national format

EXDATE Expiration date in the national format

REF. DATE Last referenced date in the national format

The QUICK format list display lists only the DATA SET NAME and the VOLUME for each data set.

The format of the printed data set list is shown in Appendix B.

After the list has been displayed, you may select items for processing by entering a one-character code to the left of the item:

B - Browse data set	C - Catalog data set
E - Edit data set	U - Uncatalog data set
D - Delete entire data set	P - Print entire data set
R - Rename entire data set	X - Print index listing
I - Data set information	M - Display member list
S - Data set information (short)	Z - Compress data set

These options are processed as follows:

- For browse, the processing is the same as if you had selected option 1 and entered the library (or data set) name on the browse entry panel, **except** that mixed mode is the assumed operation mode. If you select a library or partitioned data set, a member list is displayed.
- For edit, the processing is the same as if you had selected option 2 and entered the library (or data set) name on the edit entry panel, **except** that mixed mode is the assumed operation mode. If you select a library or partitioned data set, a member list is displayed.
- For rename, delete, catalog, uncatalog, display data set information (regular or short form), or compress, the processing is the same as though you had selected the same option from the library utility (option 3.1) or data set utility (option 3.2). For rename, a secondary panel is displayed to allow you to specify the new name. For delete, a confirmation panel is displayed to allow you to continue or cancel the operation.
- For print or print index listing, the processing is the same as though you had selected option L (print entire data set or file) or option X (print index) from the library utility (option 3.1).
- For display member list, the processing is the same as though you had displayed a member list from the library utility (option 3.1). From the member list, you may browse, print, rename, or delete selected members.

The commands available when you are displaying the data set list are:

SORT maj min Allows you to sort the list using one of the following arguments. Optionally, you can sort on a second argument by specifying the second argument after the first argument on the SORT command.

NAME alphabetic order by name
VOLUME alphameric order by volume serial
SIZE order of size, largest first
EDATE order of expiration date, earliest first
RDATE order of last referenced data, last referenced first
CDATE order of creation date, last created first
USED order of percent used, greatest first

LOCATE lparm Scrolls the list of data sets based on the field on which the data set list is sorted, as described above. If the field is not found, the list is displayed starting with the entry before which the specified field would have occurred. If the list is sorted by name, the LOCATE command specifies a name; otherwise,

Data Set List Utility (Option 3.4)

the LOCATE command specifies a creation date, size, or volume.

CONFIRM ON|OFF Controls display of the delete data set confirmation panel.

If you choose option V (display VTOC information only) or PV (print VTOC information only), you must specify the VOLUME for which you want information. The VTOC information is formatted the same, whether displayed or printed.

The fields shown on the VTOC information display or listing are:

VOLUME DATA Describes general information about the volume:

TRACKS	Total tracks on the volume
%USED	Percent of total tracks not available for allocation

VTOC DATA Describes general information about the VTOC on the volume:

TRACKS	Total tracks allocated to the VTOC
%USED	Percent of allocated tracks used by DSCBs
FREE DSCBS	Number of unused DSCBs

FREE SPACE Describes free space available for data set allocation on the volume:

TRACKS	Used for track allocation of data sets. The total number of free tracks, the number of free areas, and the largest number of contiguous free tracks are shown.
---------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

CYLINDERS	Used for cylinder allocation of data sets. The total number of free cylinders, the number of free areas with free cylinders, and the largest number of contiguous free cylinders are shown.
------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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 75). The reset utility handles only partitioned data sets with fixed or variable length records and logical record length of:

- From 10 to 255, inclusive, for fixed-length records
- From 14 to 259, inclusive, for variable-length records

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 two valid options for this utility:

- R - Reset (Create/update) ISPF statistics
- 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 - Reset (create/update) ISPF statistics
D - Delete ISPF statistics

NEW USERID          ==>          (If userid is to be changed)
NEW VERSION NUMBER  ==> 2        (If version number is to be changed)
RESET MOD LEVEL     ==> YES      (YES or NO)
RESET SEQ NUMBERS   ==> YES      (YES or NO)

ISPF LIBRARY:
PROJECT ==> SPFDEMO
GROUP   ==> XXX
TYPE    ==> COBOL
MEMBER  ==> *_                (Blank for member list, * for all members)

OTHER PARTITIONED DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>            (If not cataloged)

DATA SET PASSWORD ==>        (If password protected)
```

Figure 75. Reset Statistics Utility Panel

Reset Statistics Utility (Option 3.5)

If you have requested updating of sequence numbers, the data is scanned to determine if valid, ascending sequence numbers are present in all records. If so, the data is renumbered. Otherwise, the data is assumed to be unnumbered and renumbering is not done. If you have requested updating of the modification level, the modification level flags (the last two digits of each sequence number) are set to zeros. Otherwise, they are not changed.

The modification level and sequence numbers will not be reset unless a new version number is specified. What you specify for the RESET MOD LEVEL and RESET SEQ NUMBERS fields controls the resetting of the sequence numbers, the modification flags within the data, and the statistics. The following table shows the various combinations you can use and the results of those combinations:

RESET MOD LEVEL	YES	NO	YES	NO
RESET SEQ NUMBERS	YES	YES	NO	NO
Results:				
Sequence numbers	Reset	Reset	Unchanged	Unchanged
Modification flags	Reset	Unchanged	Unchanged	Unchanged

If the data is in packed format, there can be no sequence number processing. However, statistics for members in packed format can be created or updated if the sequence numbers are not being reset.

If you are updating the userid but not resetting the sequence numbers, the statistics are updated but the data is not scanned or renumbered.

Use the D option to delete statistics.

The version number field is required for option R and ignored for option D. 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 if requested; otherwise, unchanged
Creation date:	set to current date in the national format
Date/time last modified:	set to current date, in the national format, and time
Current number of lines:	set to the current number of data records
Initial number lines:	set to the current number of data records
Number of modified lines:	set to zero

The user id field is optional for option R; 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 non-alias members in the library. Alias members are not changed.

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.

HARDCOPY UTILITY (OPTION 3.6)

When you select this option, a panel is displayed that allows you to specify a sequential data set or member of a partitioned data set to be printed or punched, and the destination of the output (Figure 76). It also allows you to specify whether a sequential data set is to be kept or deleted after printing. (Partitioned data set members are always kept.)

Possible actions are:

- PK - Print/punch and keep data set
- PD - Print/punch and delete data set

You can specify any DASD-resident data set except the ISPF list and log data sets. Since these two data sets remain open throughout PDF processing, you can print them only upon PDF termination.

Enter a fully qualified data set name (and member name), enclosed in apostrophes. If you omit the apostrophes, your user prefix is automatically left-appended to the data set name. Note, however, that you must specify the low-level qualifier (e.g., LIST).

```

----- HARDCOPY UTILITY -----
OPTION ==> J

PK - Print/punch and keep data set
PD - Print/punch and delete data set

DATA SET NAME ==> COMPA.LIST_
VOLUME SERIAL   ==>                (If not cataloged)
DATA SET PASSWORD ==>                (If password protected PDS)

SYSOUT CLASS     ==> A
LOCAL PRINTER ID ==>

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
==> //EKMAN2  JOB (U602,B043),'EKMAN RW',NOTIFY=EKMAN
==>
==>
==>
    
```

Figure 76. Hardcopy Utility Panel

To print the data set or member using a background job, you must also specify the SYSOUT class and job statement parameters. The SYSOUT class may specify a printer or punch. Blank out the PRINTER ID field. When you have entered the desired information and press ENTER, PDF generates JCL that contains the job statement parameters and a job step that will print or punch the specified data set, using the IBM utility IEBGENER. The message "JCL GENERATED" is displayed in the short message area (line 1).

Note: IEBGENER does not support packed data. If you attempt to print packed data, the results are unpredictable. IEBGENER prints the data set one logical record per print line. If the logical record length is greater than the printer width, the logical record is truncated.

Once the JCL for the first job step is generated, the job statement parameters are shown for information only. They are no longer intensified and you may not overwrite them, since the JOB statement has already been generated.

You may then select another data set name to cause another job step to be generated. After you have entered all the data sets to be printed or punched in this job, enter the END command to cause the generated JCL to be submitted for execution, or enter CANCEL in the command/option field to exit without submitting a job.

PDF invokes the TSO SUBMIT command to accomplish job submission. The SUBMIT command displays a "JOB xxxxxxxx SUBMITTED" message accompanied by three asterisks. When you press ENTER or any other interrupt key, the previous panel is redisplayed.

If you are using the same LIST data set for multiple job steps, you should be aware that the DCB information can differ between the language processors and the linkage editor, causing an I/O error when attempting to read the the LIST data set. We suggest that you use a different LIST ID for each job step.

To route a data set or member to a local printer, enter PK or PD in the option field, then enter the data set name, volume serial (if data set is uncataloged), and password (if data set is password protected). Then enter the printer id of the local printer (assigned by your installation management) and enter blanks in the SYSOUT field. The job statement information is ignored. ISPF invokes the "DSPRINT" TSO command processor to print the data set on the specified local printer. A message is displayed in the short message area (line 1) to indicate that "DSPRINT" has accepted the request. At this point, you can specify another option or enter the END command to return to the previous panel.

Note: DSPRINT is an IBM Field Developed Program that your installation must have installed if you wish to route your data set or member to a local printer.

) LIST VTOC UTILITY (OPTION 3.7)

When you select this option, a panel is displayed that allows you to specify a volume serial for a DASD volume and, optionally, a project name or other high-level data set name qualifier (Figure 77). Possible actions are:

P - Print VTOC entries
blank - Display VTOC entries

The specified volume must be mounted unless it is a virtual volume on an IBM 3850 Mass Storage System.

If you select option P (print VTOC entries), a VTOC listing for the designated volume is recorded in the ISPF list file.

If you leave the option field blank, the VTOC listing is displayed at the terminal in browse mode. You can scroll the listing up and down using the scrolling commands or the LOCATE command. You can enter all browse commands from this display. To terminate the listing, enter the END command, which causes a return to the VTOC utility panel.

```

----- VTOC UTILITY -----
OPTION ==>

P - Print VTOC entries
blank - Display VTOC entries

VOLUME ==> 123456_
PROJECT ==>          (Blank for entire volume)

```

Figure 77. List VTOC Utility Panel

For either option, you can limit the VTOC listing to data sets that begin with a designated high-level qualifier (specified by the project name on the list VTOC selection panel).

This option can also provide information about indexed VTOCs, generated by the Data Facility/Device Support for OS/VS2 MVS Program Product, Program Number 5740-AM7.

A sample VTOC display is shown in Figure 78.

```

VTOC LISTING FOR VOLUME D00163 ----- LINE 000000 COL 001 080
COMMAND ==>                               SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-***
VOLUME: D00163
UNIT: 3330

VOLUME DATA:          VTOC DATA:          FREE SPACE:  TRACKS  CYLINDERS
   TRACKS:      7676      TRACKS:          18      SIZE:      3778      176
   %USED:        52 %     %USED:        39 %     LARGEST:   2372      124
   DATA SETS:   251      FREE DSCBS:   435
   TRKS/CYL:    19
                                     FREE EXTENTS:  105

DATA SET NAME                DSORG   TRACKS   %USED   XTENTS
-----
SILVA.COMP1.LIST             PS       7 TRKS   85 %    1 X
SILVA.DEMO.CLIST             PS       1 TRKS  100 %    1 X
SPFBACK.PID1.HELP            PO       3 TRKS   33 %    1 X
SPFBACK.PID1.JCL             PO       4 TRKS   50 %    1 X
SPFBACK.PID1.LEL             PO      10 TRKS   20 %    1 X
SPFBACK.PID1.LOAD            PO      33 TRKS   72 %    1 X
SPFBACK.PID1.MACROS          PO       2 TRKS  100 %    1 X
SPFBACK.PID1.MENUS           PO     125 TRKS   80 %    2 X
SPFBACK.PID1.MSGS            PO      17 TRKS   58 %    1 X
SPFBACK.PID1.OBJ             PO      34 TRKS   85 %    1 X
SPFBACK.PID1.PLS             PO     342 TRKS   80 %    5 X

```

Figure 78. Display VTOC Example

Track values do not include the remaining alternate tracks for the volume. The free space track values are the number of tracks for the free cylinders plus any additional free tracks.

OUTLIST UTILITY (OPTION 3.8)

This utility gives you the ability to browse, print, delete, or requeue job output that is in a held SYSOUT queue. When you select this option, a panel is displayed that allows you to select an action and enter the appropriate parameters (Figure 79). Possible actions are:

- L - List job names via the TSO STATUS command
- D - Delete job output
- P - Print job output
- R - Requeue job output to a new output class
- blank - Display job output

The jobname parameter specifies the held SYSOUT job. It is required for all options except option L. The class parameter specifies the SYSOUT hold queue. If you omit the class parameter, all SYSOUT queues are searched for the specified job. The jobid parameter is required only if more than one job exists with the same job name.

```

----- OUTLIST UTILITY -----
OPTION ==> _

L - List job names/id's via the TSO STATUS command
D - Delete job output from SYSOUT hold queue
P - Print job output and delete from SYSOUT hold queue
R - Requeue job output to a new output class
blank - Display job output

FOR JOB TO BE SELECTED:
JOBNAME ==> KRAMERA
CLASS ==>
JOBID ==>

FOR JOB TO BE REQUEUED:
NEW OUTPUT CLASS ==>

FOR JOB TO BE PRINTED:
PRINTER CARRIAGE CONTROL ==> (A for ANSI )
(M for machine )
(blank for none)
    
```

Figure 79. Outlist Utility Panel

If you select option L (list job names), a list of job names and job id's is displayed. If you supply no job name, or if the job name is your id plus one character, the status is displayed for all jobs that begin with your id. If you supply any other job name, the status for that exact job is displayed.

The list of job names is displayed on the lower portion of the panel. If the list is too long to fit on the screen, three asterisks are displayed on the last line of the screen. You can display the remainder of the list by pressing ENTER.

If you select option D (delete job output), the held output for a specific job is deleted from the specified SYSOUT queue.

If you select option P (print job output), the held output for a specific job is removed from the SYSOUT queue and placed in an ISPF-defined data set for printing. You can choose the record format for this data set by putting an entry in the "PRINTER CARRIAGE CONTROL" field. A indicates that the data contains ANSI carriage control characters, M indicates that the data contains machine control characters, and a blank field indicates that the data contains no carriage control characters. The record formats for the corresponding data sets are FBA, FBM, and FB, respectively. After you press ENTER, a secondary panel (Figure 80) is displayed on which you can enter information about how and where the job output is to be printed.

If you select option R (requeue job output), the held output for a specific job is requeued to another SYSOUT class from the specified SYSOUT queue. You must enter the new SYSOUT class on the menu.

If you leave the option field blank (display job output), the held output for the specified job is displayed in browse mode. You can use all browse commands. The data remains in the SYSOUT queue. When you enter the END or RETURN command to terminate browse, the outlist panel is redisplayed and you may then choose to print, requeue, or delete the job output.

----- OUTLIST UTILITY PRINT OPTIONS -----

OPTION ==> PD_

PK - Print data set and keep	K - Keep data set (without printing)
PD - Print data set and delete	D - Delete data set (without printing)

If END command is entered, data set is kept without printing.

DATASET NAME:

SYSOUT CLASS ==> A	(For system printer)
PRINTER ID ==>	(For 328x printer)

JOB STATEMENT INFORMATION: (Required for system printer)

==> //KRAMERA JOB (U602,BG43),MHK

==>

==>

==>

Figure 80. Outlist Print Panel

COMMAND TABLE UTILITY (OPTION 3.9)

The command table utility allows you to create or modify application command tables. When you select this option, a panel (Figure 81) 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.

Notes:

1. 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. If you use the TSO COPY command, specify the NONUM parameter. Otherwise, you will have an unusable command table.
2. While this utility is being used to modify a command table, the table may not be used for other purposes. For example, you may not use split screen and select a function with NEWAPPL(XYZ), if you are modifying command table XYZCMDS.

```
----- 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 81. Command Table Utility Panel

```
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 82. Command Table Editing Panel

The named table is displayed on an edit panel (Figure 82). 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.

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 83) allows you to specify whether you are converting menus or messages (option 1 or 2) and to specify the "from" library, which must be a partitioned data set. The second panel (Figure 84) allows you to specify the "to" library, which must also be a partitioned data set.

Note: Old-format members are limited to a maximum of 180 input records.

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.

Note: Members in the "from" data set may be in packed or unpacked format, but data in the "to" data set will always be in the unpacked format.

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" DATA SET BELOW.
THEN PRESS ENTER KEY TO SPECIFY "NEW FORMAT" DATA SET.

FROM ISPF LIBRARY:
PROJECT ==> SPF22
GROUP   ==> OURMODS
TYPE    ==> MENUS
MEMBER  ==>                (Blank for member list, * for all members)

FROM OTHER PARTITIONED DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>                (If not cataloged)

DATA SET PASSWORD ==>                (If password protected)

```

Figure 83. "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 for MVS 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" DATA SET BELOW.

TO ISPF LIBRARY:

PROJECT ==> ISP
GROUP ==> OURMODS
TYPE ==> ISPLIB

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:

DATA SET NAME ==>
VOLUME SERIAL ==> (If not cataloged)

DATA SET PASSWORD ==> (If password protected)

REPLACE LIKE-NAMED MEMBERS ==> (YES or NO)

Figure 84. "To" Panel for Converting Menu/Message Definitions

FORMAT DEFINITION UTILITY (OPTION 3.11)

The format utility is provided for support of the IBM 5550 terminal using the Double-Byte Character Set (DBCS). It is used to maintain formats that are used when editing and browsing to display data sets containing predefined format records.

The purpose of a format is to structure data from a record into fields and to define the order in which these fields are to be physically displayed on the screen when you are editing and browsing.

When you select this option, a panel (Figure 85) is displayed that lets you:

- A - Add a new format
- C - Copy formats
- D - Delete a format
- U - Update a format
- L or blank - Display the format list

----- FORMAT SPECIFICATION -----

OPTION ==>

- A - Add a new format
- C - Copy formats
- D - Delete a format
- U - Update a format
- L or BLANK - Display format list

FORMAT NAME ==> _

FOR COPY OPERATIONS, SPECIFY THE FOLLOWING:

FROM FORMAT ==> (Blank for format list, * for all formats)

FROM TABLE ==> (Default is "ISRFORM")

NOTE: The Format Utility is provided for support of the IBM 5550 terminal using the Double-Byte Character Set (DBCS).

Figure 85. Format Specification Panel

Format Definition Utility (Option 3.11)

If you specify option A and a format name, the format definition panel (Figure 86) is displayed. You can then define up to 20 fields. A field definition includes:

- The field number
- The starting column position in the record
- The field length (in bytes - maximum 71 bytes)
- The field type (EBCDIC, DBCS, or mixed data).

Note: All three of these field types may contain extended graphics characters. CAPS ON processing is not possible due to context dependencies; therefore, it is ignored when you are editing with a format.

The format definition information applies to both existing records and inserted records in a data set.

Note: It is recommended that you not use formats with numbered data, either STD or COBOL; if you do, the results are unpredictable. If you must use numbered data, either leave the columns in which the sequence numbers will appear undefined, or define an EBCDIC or mixed data field for them.

----- FORMAT DEFINITION (FORM01) -----

COMMAND ==>

FIELD NUMBER	START COLUMN	FIELD LENGTH	FIELD TYPE	FIELD NUMBER	START COLUMN	FIELD LENGTH	FIELD TYPE
1	0	0	-	2	0	0	-
3	0	0	-	4	0	0	-
5	0	0	-	6	0	0	-
7	0	0	-	8	0	0	-
9	0	0	-	10	0	0	-
11	0	0	-	12	0	0	-
13	0	0	-	14	0	0	-
15	0	0	-	16	0	0	-
17	0	0	-	18	0	0	-
19	0	0	-	20	0	0	-

FIELD NUMBER: Identifies the field position on the screen.

START COLUMN: Specifies the column position in the record.

FIELD LENGTH: From 1 to 71; fields should not overlap.

FIELD TYPE : E - Single-byte code, D - Double-byte code, M - Mixed data

Enter the END command to exit and save the format.

Enter the CANCEL command to exit without saving the format.

Figure 86. Format Definition Panel

If you specify option C in the format specification panel, the following occurs:

- If you specify both an '*' in the 'from format' field, and a table name other than ISRFORM in the 'from table' field, all formats cataloged in the 'from table' are copied to ISRFORM.

Note: If you specify a table name in the 'from table' field, and that table does not have the same format as that of ISRFORM, a severe error occurs.

- If you specify both a format name and a 'from format', the format is copied. If you specified a 'from table' (other than ISRFORM), the format is copied from that table. Otherwise, the format is copied from ISRFORM. The format definition panel for the newly-created format, containing the currently-defined fields, is displayed. You can add, delete, and/or update field definitions. When you press the END key, the format definition is cataloged in ISRFORM under the format name you specified.
- If you specify a format name but no 'from format', the copy format selection list panel (Figure 87) is displayed.

Format Definition Utility (Option 3.11)

If you did not specify a 'from table', the formats listed are those cataloged in ISRFORM, the default format table. Otherwise, the formats listed are those cataloged in the table you specified.

You can select a format to copy by entering 'S' to the left of that format name. The format is copied, and the format definition panel for the newly-created format, containing the currently-defined fields, is displayed. You can add, delete, and/or update field definitions. When you press the END key, the format definition is cataloged in ISRFORM under the format name you specified in the format specification panel.

```
----- COPY FORMAT SELECTION LIST (COMMON) -----
COMMAND ==>                                     SCROLL ==>

  NAME      RENAME      CREATED  LAST MODIFIED  ID
S  COMMON1          05/02/83 05/06/83 10:13:00  Z72KXX
   COMMON2          05/02/83 05/06/83 10:14:23  Z72KXX
   COMMON3          05/02/83 05/06/83 10:14:18  Z72KXX
S  COMMON4          05/02/83 05/06/83 10:16:01  Z72KXX
```

Figure 87. Copy Format Selection List Panel

- If you specify neither a format name nor a 'from format', but you do specify a 'from table' (other than ISRFORM), the copy format selection list panel is displayed. You can select one or more formats to copy by entering 'S' to the left of the format name(s). Each of these formats is copied under the same name from the specified 'from table' to the ISRFORM table.

Note: If you do not specify option C but specify a name in the 'from format' field, the 'from format' field is ignored.

If you specify option D and a format name in the format specification panel, the format will be deleted.

If you specify option U and a format name in the format specification panel, the format definition panel containing the currently-defined fields is displayed. You can add, delete, and/or update field definitions.

If you specify option L or leave the option line blank in the format specification panel, the format selection list panel (Figure 88) is displayed. You can update, rename, or delete formats from this panel by entering 'U' (Update), 'S' (Select -- functionally equivalent to Update), 'R' (Rename), or 'D' (Delete) to the left of the entry.

```

----- FORMAT SELECTION LIST -----
COMMAND ==>                                SCROLL ==>

```

	NAME	RENAME	CREATED	LAST MODIFIED	ID
U	FORM01		05/02/83	05/06/83 10:13:00	Z72KXX
	FORM02		05/02/83	05/06/83 10:14:23	Z72KXX
	FORM03		05/02/83	05/06/83 10:14:18	Z72KXX
	FORM04		05/02/83	05/06/83 10:16:01	Z72KXX

Figure 88. Format Selection List Panel

If you specify 'U' or 'S' for a format, the format definition panel containing the currently-defined fields is displayed. You can add, delete, and/or update field definitions.

The SELECT command provides you with another way to specify a format. To do this, enter the word 'SELECT' followed by the format name on the command line. If the format exists, the format definition panel containing the currently-defined fields is displayed. You can add, delete, and/or update field definitions. If no format exists for that name, a new format is created, and the format definition panel is displayed to allow you to define fields.

If you specify 'R' for a format, you must also specify its new name under the RENAME column.

If you specify 'D' for a format, the format will be deleted.

You can specify that multiple operations be performed at the same time. However, if you specify 'U' or 'S' in conjunction with other line commands, any commands after the first 'U' or 'S' are ignored.

You can sort the name list on this panel by entering the SORT command on the command line. The format of the SORT command is:

```
SORT [NAME|TIME]
```

where:

```
NAME -- Sort by name
```

```
TIME -- Sort by time last modified
```

The LOCATE command is another useful tool, especially if you have a long format list. To use the LOCATE command, ensure that the list is sorted

Format Definition Utility (Option 3.11)

by name. Next, enter the word 'LOCATE' followed by the format name on the command line. If the format exists, the entry for the specified format name appears as the second line following the header lines. If the specified name is not found, the existing format name that immediately precedes the specified name appears as the first line following the header lines.

CHAPTER 8. FOREGROUND (OPTION 4)

The foreground option provides an interface to standard language processors for foreground compilation, assembly, linkage edit, and debugging of programs stored in ISPF libraries or other partitioned or sequential data sets. It also provides an interface to the Document Composition Facility (SCRIPT/VS) Program Product for formatting, browsing, and printing documents, as well as a facility for listing the parts for a source program module. The foreground selection panel is shown in Figure 89.

Member expansion uses simple language scanners to locate expansion triggers. These scanners do not have all of the sophistication of the actual language processors. Therefore, unusual code or code that does not compile may not be successfully processed by member expansion. Examples are:

- Trigger statements with comments that extend onto the next line
- Trigger statements that have compiler instructions to alter the content of the code to be included.

When the Session Manager Program Product, 5740-XE2, is installed, a line is added to the panel that allows you to enter Session Manager mode when you invoke any of the foreground processors.

MEMBER EXPANSION

You can control whether any of the source data to be compiled is stored in packed format by using the pack option of move/copy (option 3.3) or edit (option 2). If any of the members for a compilation are packed, you must fill in the field following SOURCE DATA PACKED with YES. If your source code contains packed data, it is unpacked before compilation and any included members are also unpacked in line at the line where the INCLUDE or COPY statement is found. This unpacking of included modules is dependent on an expansion trigger for each language to be processed.

When using languages that allow multiple compiles, such as VS/FORTRAN, you must put the program source terminating statement in your original, or top-level, program. This statement cannot be in an "included" member.

Compiler control statements and/or symbolic substitution are not considered during member expansion. The following table shows the languages processed by member expansion, their expansion trigger, syntax, and the input record format and the columns processed.

<u>LANGUAGE</u>	<u>EXPANSION TRIGGER</u>	<u>SYNTAX</u>	<u>INPUT COLUMNS PROCESSED</u>	
			<u>F/FB FORMAT</u>	<u>V/VB FORMAT</u>
Assembler	COPY	COPY name	1-80	NA
PL/1	%INCLUDE	%INCLUDE DDNAME(name); %INCLUDE name;	2-72	9-100
COBOL	COPY	COPY name.	7-72	NA
VS FORTRAN	INCLUDE	INCLUDE (name)	1-72	NA
PASCAL	%INCLUDE	%INCLUDE name; %INCLUDE DDNAME(name);	1-72	1-100
SCRIPT	.IM	.IM name .IM (name) .IM ('name')	1-reclength or 9-reclength	1-reclength or 1-(reclength - 8)

All languages User-trigger User-trigger name

Note that the following restrictions apply only to the member expansion and member parts listing functions:

- Expansion triggers must follow their respective language coding conventions unless otherwise noted.
- Multiple names and preprocessor variables on trigger statements are not permitted.
- User triggers and their start column are specified at installation time and must be no more than 20 characters in length and in uppercase with no imbedded blanks. No part of the user trigger may be in a comment or continuation field. In Assembler, the user trigger may not start in column 1.
- Macros may not be in packed form.
- In FORTRAN, only the fixed form of coding is permitted by the member expansion function.
- In PL/1, COBOL, and PASCAL, freeform coding is allowed except in trigger statements.
- The trigger statement must be the only statement in the logical record (card image). In other words, no continuation is allowed either into or from a trigger statement. Also, the trigger keyword must be the first nonblank character on the trigger statement.

- In SCRIPT, the '.im' statement must be the only statement in the logical record and must start in the first valid column. The first logical record is tested for line numbers, as follows:
 - For fixed-length records, if the last 8 characters are all numeric, they will be skipped for the entire library.
 - For variable-length records, if the first 8 characters are all numeric, processing will begin with column 9.

The statements may be in either uppercase, lowercase, or mixed case.

- For language processors that allow names longer than eight characters, the name will be truncated at eight. For COBOL, the name will be truncated at eight and also at the first dash (-).
- For language processors that allow uppercase and lowercase names, all referenced names are converted to uppercase.

Member Expansion Return Codes

Return Code	Explanation
0	Normal completion
n where n = 1 through 14	parm n was too long
10	Too many parameters
12	Too few parameters
20	Severe error in expand module

COBOL Restrictions

The first statement in the COBOL program must be either an expansion trigger or a valid COBOL division header. If an expansion trigger is the first statement, it must start in FIELD B.

In the COPY statement, the text-name is the only thing processed. The statement must end on the same line as the COPY keyword with a period followed by a space. If any option is found, the COPY statement will not be expanded.

In the IDENTIFICATION DIVISION, the division header or paragraph header statements must be blank except for the division or paragraph name. The trigger statement must be on the next nonblank, noncomment line.

In all other divisions, the trigger statement (line) may be on any line in the division.

If the "WITH DEBUGGING MODE" clause is not found in the SOURCE COMPUTER paragraph, all debug lines will be passed to the compiler as is. If the clause is found, valid trigger statements found on debug lines will be

expanded and a "D" will be inserted in column 7 of all noncomment noncontinuation lines included.

Any nonblank character found in FIELD A will cause the end of the paragraph form of the NOTE statement.

Trigger Statement Errors

If an error, such as restricted option, statement on more than one line, or referenced member name not found, the trigger statement is not expanded and is passed on to the language processor as-is.

In SCRIPT, if the error was found in a user trigger one blank line will be inserted before and after the statement in question.

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

  1 - System assembler           7 - Linkage editor
  2 - OS/VS COBOL compiler      9 - SCRIPT/VS
  3 - VS FORTRAN compiler     10 - COBOL interactive debug
  4 - PL/I checkout compiler   11 - FORTRAN interactive debug
  5 - PL/I optimizing compiler 12 - Member parts list
  6 - PASCAL/VS compiler

SOURCE DATA PACKED ==>          (YES or NO)
```

Figure 89. Foreground Selection Panel

PROCESSING SEQUENCE

The sequence for foreground processing is:

1. Select an option from the foreground selection panel and press ENTER.
2. An entry panel is displayed for the selected foreground processor. On this display, you may enter the appropriate data set and member names, library concatenation sequence, and processor options. Figure 90 shows an example for the assembler. Either an ISPF library or other partitioned or sequential data set must be specified. The other fields on this panel are optional. See "Assembler/Compiler Processing" for a discussion of these fields.
3. If the input data set is partitioned 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.
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."

```

----- FOREGROUND ASSEMBLY -----
COMMAND ==>

ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  GROUP   ==> XXX      ==> A          ==> MASTER   ==>
  TYPE    ==> ASM
  MEMBER  ==> TOP      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>

LIST ID ==> LISTASM_          PASSWORD ==>

ASSEMBLER OPTIONS: (Options LIB, LOAD, and PRINT generated automatically)
  ==> LIST,TEST,TERM,RENT

ADDITIONAL INPUT LIBRARIES:
  ==> 'SPFTEST.FLAG.ASM'
  ==>
  
```

Figure 90. Foreground - Assembler Example

5. Once all of the input parameters have been specified, the foreground processor is invoked. If the Session Manager is installed and if you specified Session Manager mode on the foreground selection panel, execution is under control of the Session Manager. All PF and PA keys are under Session Manager control. When foreground processing is complete, you are prompted to enter a null line to return to PDF control.

If the Session Manager is not invoked, the PA and PF keys have their usual TSO-defined meaning (generally, the PF keys are treated the same as the ENTER key). Communication with the foreground processor is in line-I/O mode. When foreground processing is complete, you will see three asterisks (***) . Press ENTER.

6. If the foreground processor generated an output listing, the listing is displayed automatically in browse mode. The output can be scrolled up or down using the scrolling commands. All the browse commands are available. When you have finished browsing the listing, enter the END command.
7. After you browse the output listing, the foreground print panel is displayed to allow you to print, keep, or delete the output (Figure 91). On this panel, the data set name field shows the name of the data set specified in the list-id field on the foreground entry panel. This data set is processed as you specify on the print panel. When you press the ENTER key, the processor entry panel is redisplayed.
8. When the entry panel is redisplayed, a message indicating normal or abnormal completion 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 list data set is retained, but the foreground print panel is not displayed.

ASSEMBLER/COMPILER PROCESSING

The assembler entry panel shown in Figure 90 is typical of the language processor panels. In the figure, a library concatenation sequence of three ISPF data sets and one additional input library has been specified. The concatenation order is:

```
SPFDEMO.XXX.ASM  
SPFDEMO.A.ASM  
SPFDEMO.MASTER.ASM  
SPFTEST.FLAG.ASM
```

```

----- FOREGROUND PRINT OPTIONS -----
OPTION ==> _

PK - Print data set and keep          K - Keep data set (without printing)
PD - Print data set and delete        D - Delete data set (without printing)

If END command is entered, data set is kept without printing.

DATA SET NAME: KIRK.LISTASM.LIST

SYSOUT CLASS ==> A                    (For system printer)
PRINTER ID   ==>                     (For 328x printer)

JOB STATEMENT INFORMATION:           (Required for system printer)
==> //KIRKM JOB (AB01,B001,123456),M.KIRK
==> //
==>
==>

```

Figure 91. Foreground Print Panel

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). If member TOP first appears in data set SPFDEMO.A.ASM, the following prompter command would be generated:

```

ASM 'SPFDEMO.A.ASM(TOP)'
LIB('SYS1.MACLIB',
    'SPFDEMO.XXX.ASM',
    'SPFDEMO.A.ASM',
    'SPFDEMO.MASTER.ASM',
    'SPFTEST.FLAG.ASM')
LOAD('SPFDEMO.XXX.OBJ(TOP)')
PRINT(LISTASM)
LIST,TEST,TERM,RENT

```

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 is passed to the prompter (via the LIB parameter) to allow inclusion of subsidiary members referenced by COPY statements in the source code.

2. The object module produced by the assembler is placed in a partitioned data set with a 3-level name composed of the project name, the first library name, and a type qualifier of OBJ. The member name for the object module is the same as the primary member to be assembled.

If the input is from another partitioned or sequential data set, the object module is written to a data set with the same name as the input data set, but the lowest level qualifier is replaced with OBJ. The OBJ data set organization (partitioned or sequential) matches that of the input data set. For partitioned data sets, the member name of the object module is the same as the input member name.

3. The list id that you specify is passed to the prompter via the PRINT parameter. In the example, the listing produced by the assembler is placed in a sequential data set named:

```
prefix.LISTASM.LIST
```

(The prompter automatically left-appends your TSO data set prefix and right-appends the LIST qualifier.)

You may omit the list id if the input is from a partitioned data set; the list id defaults to the member name. You must specify a list id if the input is from a sequential data set.

4. The assembler options are passed to the prompter exactly as you specify them.

The commands generated for the other language processors are similar to the assembler example, except that:

- The macro library 'SYS1.MACLIB' is not included in the concatenation sequence passed via the LIB parameter.
- For COBOL, an optional TEST parameter may be generated, as follows:

```
TEST ('project-name.lib1-name.SYM(member-name)')
```

You may control the generation of this parameter from the panel.

Note: If TEST is specified, you must pre-allocate space for the SYM data set. You can use the PDF data set utility to allocate the SYM data set.

- For FORTRAN, there is no LIB parameter. The concatenation sequence is still used to locate the member to be compiled.
- For the two PL/I compilers and PASCAL/VS, an OBJECT parameter (rather than LOAD) is generated by PDF to designate the destination of the object module.

- For the PL/I Checkout Compiler, the OBJECT parameter is optional (controlled by a field on the selection panel). If you specify OBJECT, the following PLIC parameter is generated by PDF:

```
OBJECT('project-name.lib1-name.OBJ(member-name)'
      'project-name.lib1-name.ITEXT')
```

If you specify NOOBJECT, neither OBJ nor ITEXT data is generated.

In addition to constructing the input data set name, the PDF foreground processing option automatically generates the following optional parameters. Accordingly, you should not specify these parameters on the options line of the entry panel.

```
For ASM:          LIB, LOAD, PRINT
For COBOL:        LIB, LOAD, PRINT, TEST
For FORT:         LOAD, PRINT
For PLIC:         LIB, OBJECT, PRINT
For PLI:          LIB, OBJECT, PRINT
For PASCAL/VIS:  LIB, OBJECT, PRINT
For LINK:         LIB, LOAD, PRINT
For TESTCOB:     LOAD, PRINT, PARM
For TESTFORT:    LIB, PRINT, SOURCE
```

LINKAGE EDITOR PROCESSING

For the linkage editor, the entry panel allows you to specify up to two system subroutine libraries (Figure 92). The concatenation sequence used by PDF to locate the member for input to the linkage editor is:

```
project-name.lib1-name.type
project-name.lib2-name.type
(etc.)
```

where "type" is whatever you specify on the panel: It may be OBJ or it may be some other type containing linkage editor control statements. If the type is not OBJ, an "OBJECT" DDNAME is automatically allocated to facilitate use of the following linkage editor control statements.

```
INCLUDE OBJECT(member-name)
```

Example:

```
PROJECT ==> XYZ
LIBRARY ==> MYLIB    ==> MASTER    ==>
TYPE     ==> LEL
MEMBER   ==> TOP
```

In this example, PDF searches data sets 'XYZ.MYLIB.LEL' and 'XYZ.MASTER.LEL' to find member TOP, which should contain linkage editor language (LEL) control statements. In addition, PDF allocates to DDNAME "OBJECT" (DISP=SHR) the following concatenated sequence of object libraries:


```
XYZ.MYLIB.OBJ
XYZ.MASTER.OBJ
```

This concatenated sequence is searched by the linkage editor if member TOP contains INCLUDE OBJECT(member-name) statements.

The concatenation sequence passed to the linkage editor via the LIB parameter has a type qualifier of LOAD and includes the user-specified system libraries, as follows:

```
LIB('project-name.lib1-name.LOAD',
    'project-name.lib2-name.LOAD',
    etc.,
    'syslib1-name',
    etc.)
```

This concatenation sequence is used by the linkage editor to resolve "automatic call" references.

```
----- FOREGROUND LINKAGE EDIT -----
COMMAND ==>

ISPF LIBRARY:
PROJECT ==> SPFDemo
GROUP   ==> MYLIB   ==> MASTER   ==>
TYPE    ==> OBJ
MEMBER  ==> MAINPROG      (Blank for member selection list)

OTHER PARTITIONED DATA SET:
DATA SET NAME ==>

LIST ID ==> LLLL_          PASSWORD ==>

LINKAGE EDITOR OPTIONS:      (Options LOAD and PRINT generated automatically)
==> MAP,LET,TEST,RENT,DCBS(3072)

ADDITIONAL INPUT LIBRARIES:  (LOAD LIBRARIES ONLY)
==>
==>
```

Figure 92. Foreground - Linkage Editor Example

SCRIPT/VS PROCESSING

Selecting the SCRIPT/VS option displays a sequence of panels that allow you to do the following:

- Create SCRIPT/VS parameters
- Create styles (for SCRIPT options and print characteristics)
- Browse formatted output
- Print formatted output

Use of this facility requires the installation of the Document Composition Facility (SCRIPT/VS) program product, with the Foreground Environment feature.

The first panel displayed allows you to specify the following:

- Formatting style (the set of SCRIPT/VS options)
- ISPF library member
- Other partitioned or sequential data set
- Option to browse the scripted output and (optionally) the output listing file ID

```

----- SCRIPT/VS SELECTION MENU -----
COMMAND ==>

STYLE ==>                                DISPLAY STYLE OPTIONS ==> NO      (Yes or No)

ISPF LIBRARY:
  PROJECT ==> ISPFDEMO
  GROUP   ==> xxx      ==> TEST      ==> MASTER ==>
  TYPE    ==> TEXT
  MEMBER  ==> TOP_      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>

DATA SET PASSWORD ==>                    (If password protected)

LIST ID   ==>                            (Blank defaults to member name)

BROWSE    ==> YES                        (Yes or No)
  
```

Figure 93. SCRIPT/VS Utility - Selection Panel

If you do not define any formatting styles, SCRIPT/VS creates two styles for you. They are DRAFT and FINAL. These two styles correspond to the formatting options available in the previous release of SCRIPT/VS Foreground Processing. If you have not defined these options before, or

if this is your first release of ISPF/PDF, the default values for the SCRIPT/VS formatting options are set for you.

If you leave the style field blank, a panel displays a list of the available styles. If you enter the style name of an existing style in the style field, the existing style is used for formatting. If you enter a new style name, the name is added to your style list. The new style uses SCRIPT/VS formatting options that are equal to the formatting options of the last style.

For information on specifying the ISPF library member or other partitioned or sequential data sets, see Chapter 3, "Libraries and Data Sets" on page 25. You can specify concatenation of up to four groups on the SCRIPT/VS Formatting Style Selection Panel.

The formatted listing is stored in a data set named:

prefix.list-id.LIST

where list-id can be specified on the SCRIPT/VS Selection Panel and prefix is the TSO data set prefix.

If you leave the list-id field blank and the input is from a partitioned data set, the list-id is the same as the member name. If the input is from a sequential data set, you must specify the list-id. This data set will be used again for each invocation.

```
----- SELECT SCRIPT/VS FORMATTING STYLE -----
COMMAND ==>                                     SCROLL ==> CSR

Valid Line Commands:  S - Use this STYLE for formatting
                      D - Delete

                                DISPLAY FORMATTING OPTIONS ==>

Line
Cmd  Style      Description
1'   DRAFT      Draft document SCRIPT/VS formatting options
1'   FINAL      Final document SCRIPT/VS formatting options
S    IMEMO      Format options for inter-office memos
```

Figure 94. SCRIPT/VS Formatting Style Selection Panel

The SCRIPT/VS Formatting Style Selection Panel is a scrollable list that contains all the styles available to you. Each style is a set of predefined formatting options. You can use two line commands on this panel:

S Select this style for formatting

D Delete this style from the list of available styles

The description area indicates the purpose of each style. You can modify this area.

You can only select one style at a time for formatting; however, one or more styles can also be deleted at the same time. If you chose to display the SCRIPT/VS formatting options, the SCRIPT/VS specification panel is displayed here. Otherwise, formatting of the SCRIPT/VS document will start.

```

----- SCRIPT/VS OPTIONS FOR STYLE: IMEMO -----
COMMAND ==>

PROFILE DATA SET ==>
FONTLIB DATA SET ==>
SEGLIB DATA SET ==>
MACRO DATA SET ==>

BIND:      Odd ==>
           Even ==>

DEVICE TYPE ==>
CHARS (Fonts) ==>
SYSVAR ==>
PAGE ==>

Enter Yes or No for the following:
Twopass ==>
Spelling Check ==>
Index ==>
Line Number ==>

Uppercase only ==>
Unformat ==>
Condensed text (CTF) ==>
    
```

Figure 95. SCRIPT/VS Options Specification Panel

The document options specification panel allows you to specify the SCRIPT/VS options for formatting the ISPF library member of other partitioned or sequential data sets. All parameters on this panel are optional. For a complete description of the SCRIPT/VS options, see the Document Composition Facility: User's Guide, SH20-9161.

The options are as follows:

- PROFILE ID** Enter the data set name that contains the profile to be imbedded before the primary file.
- FONTLIB** Enter the name of a fully qualified partitioned data set or sequential data set that contains the font library.
- SEGLIB** Enter the name of a fully qualified partitioned data set or sequential data set that contains the page segment.
- MACRO** Enter the data set name or library to be searched for MACRO and style definitions.
- BIND** Enter values to shift the formatted output of each page to the right for binding purposes. If you do not specify an even value, the odd value is used for the even pages. Binding values can be any valid space units. For example, a value of 4 means 4 characters, a value of 2i means 2 inches, and a value of 6cm means 6 centimeters.
- DEVICE TYPE** Enter a logical output device type (for example, 1403W8 or 3800N12S).
- CHARS (Fonts)** Enter all fonts to be used in formatting the document. The first font you specify is the initial font.
- SYSVAR** Enter values to be set in the symbols &SYSVARn in the following form:
- n value
- where n is the single-character designation of the system variable you are specifying, and value is the value you are assigning to the variable.
- PAGE** Enter selected pages to format and print.

To print these pages	Enter these values
4, 5, and 6	'from 4 to 6'
9 and 10	'from 9 for 2'

Figure 96 (Part 1 of 2). Example of specifying specific pages to print from a document

To print these pages	Enter these values
14	'from 14 only'
2, 3, 4, 10, 12, 13, and 14	'from 2 to 4 from 10 only from 12 for 3'

Figure 96 (Part 2 of 2). Example of specifying specific pages to print from a document

For the following options you should specify YES, NO, or leave the field blank. If you leave the field blank, it is the same as specifying NO.

- TWOPASS** Enter YES to process the document twice. Formatted copy is produced on only the second pass.
- SPELLING CHECK** Enter YES to enable the .SV (Spelling Verification) control word.
- INDEX** Enter YES to enable the .PI (Put Index) control word.
- LINE NUMBER** Enter YES to print the file ID and line number of the last line read to the right of each formatted line.
- UPPERCASE ONLY** Enter YES to change all lowercase letters to uppercase letters before printing.
- UNFORMAT** Enter YES to print all input lines without formatting.
- CONDENSED TEXT (CTF)** Enter YES to prepare the document as input for the STAIRS/VS program product. If you specify YES for this option, the device type must be STAIRS or blank; otherwise, it will be ignored. The output will be placed in data set

prefix.DSMUTCTF.TEXT

where prefix is the TSO data set prefix.

If you enter the END command from the SCRIPT/VS Formatting options panel, changes to the formatting options will not be saved, except for the follow exception: if the style is new, it is saved with default formatting options. Pressing the ENTER key while this panel is displayed formats the document.

When the SCRIPT/VS processor finishes you can optionally browse the formatted document.

After you browse the formatted document, the SCRIPT/VS print panel is displayed. The print panel gives you the option of printing the formatted document and specifying its disposition. Your options are:

- PK Print the file and retain it.
- PD Print the file, then delete it.
- K Retain the file.
- D Delete the file and do not print it.

MEMBER PARTS LIST

The member parts list shows, for each source program module specified, the following information:

- The names of the modules it calls or includes
- The names of the modules that call or include it

The languages permitted in the member expansion function are also permitted in the member parts list function and the expansion triggers have the same restrictions. In addition to the expansion triggers, the member parts list also uses the CALL statements in Assembler, PL/I, COBOL, and VS FORTRAN. The format of the CALL statement is "CALL name" where the delimiter after the name can be either a left parenthesis, a blank, or a valid statement delimiter. In COBOL, the CALL statement is only valid in the PROCEDURE DIVISION, and the statements CALL PGMA and CALL 'PGMA' both result in a reference to the member name PGMA.

When you select the member parts list option from the PDF foreground option panel, the panel shown in Figure 97 is displayed.

```

----- FOREGROUND MEMBER PARTS LIST -----
OPTION ==>

  1 - Print member parts
  2 - Write member parts data set
  3 - Browse member parts

ISPF LIBRARY:
PROJECT ==>
GROUP ==>          ==>          ==>          ==>
TYPE ==>
MEMBER ==>          (Blank for member list, * for all members)

LANGUAGE ==>          (Defaults to type value)

GROUPS FOR PRIMARY MEMBERS ==> (1, 2, 3, or 4)

SYSOUT CLASS ==>      (Defaults to 'A')

OUTPUT DATA SET:          (Option 2 only)
  DATA SET NAME ==>
  
```

Figure 97. Foreground Member Parts List Panel

Note: The SOURCE DATA PACKED field on the foreground selection panel is not used by member parts list; both packed and unpacked data sets can be read.

On the foreground member parts list panel, enter the appropriate ISPF library and member names, group concatenation sequence, and listing options.

You must specify one of the processing options, the library and member name, the source language, and the groups for primary members.

A member name may be a unique name, or a truncated name, or an asterisk (*). A truncated name is specified by giving a member name prefix followed by an asterisk (for example, ISRPR*). If the member name is an asterisk (*), all members from the primary libraries are used as primary members. A member name of blank results in a member list being displayed; you may select only one member from this list.

The print option, option 1, prints the member parts listing 120 characters wide.

Note: The job cards from option 5 are used to submit the print job.

The data set option, option 2, produces an intermediate sequential member parts list in the user-named data set. This data set may be either a sequential data set or a member of a partitioned data set. If the data set has not been allocated, option 2 allocates it with an LRECL of 17, a BLKSIZE of 3009, and a RECFM of FB. The format of the records in the user-named data set is as follows:

<u>Field Name</u>	<u>Format</u>	<u>Description</u>
Member name	CHAR(8)	Subject member
Called by or calls member name	CHAR(8)	Referenced member
Call flag	BIT(1)	Found on a CALL statement
Include flag	BIT(1)	Found by INCLUDE or COPY
Not found flag	BIT(1)	Referenced member not found
From flag	BIT(1)	Subject called from referenced member
To flag	BIT(1)	Referenced member called from subject member
COBOL flag	BIT(1)	Member referenced outside valid COBOL division
Unused	BIT(2)	

The display option, option 3, allows you to browse the member parts list. Figure 98 shows an example of a member parts list display where:

- Library ISPFPROJ.ABL.PLI contains members:

- MEMBERA - has no calls or includes
- MEMBERB - calls MEMBERC
- MEMBERC - calls MEMBERD and MEMBERG, and includes MEMBERE

- Library ISPFPROJ.STAGE.PLI contains members:
 - MEMBERD - includes MEMBERE
 - MEMBERE - has no calls or includes
 - MEMBERF - has no calls or includes
- Library ISPFPROJ.STAGE.PLI is concatenated to library ISPF.ABL.PLI.
- A parts list is requested for all members in the first data set.

BROWSE - PARTS LIST FOR ISPFPROJ.ABL.PLI(*)
 COMMAND ==> SCROLL ==> PAGE

FROM	VIA	FROM	VIA	MEMBER	TO	VIA	TO	VIA
-----	---	-----	---	-----	-----	---	-----	---
***** TOP OF DATA *****								
				(MEMBERA)				
				(MEMBERB)	MEMBERC	C		
		MEMBERB	C	(MEMBERC)	MEMBERD	C	MEMBERE	I
					MEMBERG	C*		
		MEMBERC	C	(MEMBERD)	MEMBERE	I		
MEMBERC	I	MEMBERD	I	(MEMBERE)				
***** BOTTOM OF DATA *****								

Figure 98. Member Parts List Display

The source language must be either ASSEMBLER (ASM), COBOL (COB), FORTRAN (FOR), PL/1 (PLI or PL1), PASCAL (PAS), or SCRIPT (SCR).

The GROUPS FOR PRIMARY MEMBERS field is used to identify which concatenated libraries are to be used in locating primary members. A primary member is a member that starts the member parts expansion chain. The chain is broken when a member cannot be found in the set of concatenated libraries or there are no more unique members referenced. If a member cannot be found, it is flagged on the listing. For instance, internally called routines are not found.

A GROUPS FOR PRIMARY MEMBERS value of "1" specifies that only the first library in the concatenation sequence is to be used in locating primary members. A value of "2" specifies that the first and second libraries are to be used and so forth. These specified libraries are known as primary libraries.

Note: If a member cannot be found, the name is flagged with an asterisk (*) and processing continues. When no more primary members can be found, the listing is printed or browsed. Calls to internal routines or variable names will result in the 'member not found' flag being set.

CHAPTER 9. BATCH (OPTION 5)

The batch option provides an interface with standard language processors for background compilation, assembly, or linkage edit of programs stored in ISPF libraries or other partitioned or sequential data sets, as well as providing a way of listing the component parts of a source program module.

The batch selection panel is shown in Figure 99. It allows you to enter job statement information and to select a language processor or select the member parts listing function.

The SOURCE DATA ONLINE field allows you to specify whether the data to be processed resides on a currently mounted volume. If you specify "yes," PDF checks the data set information that you entered on the batch panel, and allows you to display a member list. If you specify "no," PDF assumes that the data cannot be accessed except by the batch job, and does not verify the existence or validity of the specified data set.

```

----- BATCH SELECTION MENU -----
OPTION ==> _

    1 - System assembler           5 - PL/I optimizing compiler
    2 - OS/VS COBOL compiler       6 - PASCAL/VS compiler
    3 - VS FORTRAN compiler        7 - Linkage editor
    4 - PL/I checkout compiler     12 - Member parts list

SOURCE DATA ONLINE ==> YES (YES or NO)
SOURCE DATA PACKED ==> YES (YES or NO)

JOB STATEMENT INFORMATION: (Verify before proceeding)
==> //SILVA7  JOB (U602,B043),'SILVA RJ',CLASS=B,
==> //          MSGLEVEL=2,
==> //          NOTIFY=SILVA
==>

```

Figure 99. Batch Selection Panel

The SOURCE DATA PACKED field allows you to specify whether any of the data to be used by the batch job is stored in packed format. If you specify YES, the required members will be expanded onto a temporary file which is then used as input to the batch job. For additional information about member expansion, see Chapter 8, "Foreground (Option 4)" on page 229.

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. 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 or link edits within the same job.

The data entry panels for batch processing are similar to those for foreground, except that you may enter either a list id (if the output listing is to be stored in a data set) or SYSOUT class (if the listing is to be printed as part of the batch job). If you specify both a list id and SYSOUT class, the list id is used. An example for the PL/I Optimizing Compiler is shown in Figure 100.

```
----- BATCH PL/I OPTIMIZING COMPILE -----
COMMAND ==>

ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  GROUP   ==> MYLIB   ==> MASTER   ==> FLAG     ==>
  TYPE    ==> PLI
  MEMBER  ==> TOPSEG_      (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>

LIST ID ==>                (Blank for hardcopy listing)
SYSOUT CLASS ==> A         (If hardcopy requested)

COMPILER OPTIONS:
  ==> MACRO,XREF

ADDITIONAL INPUT LIBRARIES:
  ==> 'OURSYS.PLIMACS'
  ==>
```

Figure 100. Batch - PL/I Optimizing Compiler Example

After you fill in the entry panel and press the ENTER key, PDF generates the appropriate JCL statements. The JCL that would be generated for the PL/I example (Figure 100) is:

```
//SCAN      EXEC  PGM=ISRSCAN,PARM='TOPSEG',COND=(12,LE)
//IN        DD   DSN=SPFDEMO.MYLIB.PLI,DISP=SHR
//          DD   DSN=SPFDEMO.MASTER.PLI,DISP=SHR
//          DD   DSN=SPFDEMO.FLAG.PLI,DISP=SHR
//OUT       DD   UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(2,2)),
//          DSN=&TEMP1
//*
//PLIO      EXEC  PGM=IELOAA,REGION=100K,COND=(12,LE),
//          PARM='MACRO,XREF'
//SYSPRINT  DD   SYSOUT=(A)
//SYSIN     DD   DSN=&TEMP1,DISP=(OLD,DELETE)
//SYSUT1    DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSLIB    DD   DSN=SPFDEMO.MYLIB.PLI,DISP=SHR
//          DD   DSN=SPFDEMO.MASTER.PLI,DISP=SHR
//          DD   DSN=SPFDEMO.FLAG.PLI,DISP=SHR
//          DD   DSN=OURSYS.PLIMACS,DISP=SHR
//SYSLIN    DD   DSN=SPFDEMO.MYLIB.OBJ(TOPSEG),DISP=OLD
```

The first step executes the scan program (distributed as part of PDF), ISRSCAN (to copy one member) or ISRLEMX (to copy the primary member and expand any included members, unpacking any packed members), which searches the user-specified sequence of concatenated libraries to find the designated member. If the scan program finds the member, it copies the member to a temporary sequential data set named &TEMP1 and exits with a return code of zero. If any errors are encountered, the scan program exits with one of the following return codes, which prevent execution of the second job step:

ISRSCAN

```
12 - Member not found
16 - OPEN error on DDNAME=IN
20 - I/O error on DDNAME=IN
24 - OPEN error on DDNAME=OUT
28 - I/O error on DDNAME=OUT
```

ISRLEMX

```
1 - 9 - Parameter n was too long, where n is 1-9
10 - Too many parameters
20 - Program error - an error message should be printed
    in the ISRLMSG file
```

The second step invokes the PL/I compiler with &&TEMP1 named as the input data set. The concatenation sequence is passed to the compiler via SYSLIB DD statements, to allow inclusion of subsidiary members referenced by %INCLUDE statements in the source text. The object module is directed to a partitioned data set with a three-level name composed of the project name, the first library name, and a type qualifier of OBJ. The member name for the object module is the same as the primary member to be compiled. The compiler listing is directed to SYSOUT class A, as specified.

For batch assembly and link edit, an optional SYSTEM DD statement is also generated (if you specify TERM), as follows:

```
//SYSTEM DD DSN=prefix.member.TERM,DISP=(MOD,CATLG)
```

where "prefix" is the TSO user prefix, and "member" is the same member name as specified on the entry panel.

When PDF has generated the JCL, it returns to the batch selection panel (unless you bypassed that selection 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 JCL and returns directly to the primary option menu. In either case, you may exit from the entry panel without generating any JCL 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 overtyped, since the JOB statement has already been generated. At this point, you may select another processor to cause more JCL to be generated, cancel by entering the option CANCEL, or enter the END or RETURN command to cause the generated JCL to be submitted for execution.

PDF invokes the TSO SUBMIT command to accomplish job submission. The SUBMIT command displays a "JOB xxxxxxxx SUBMITTED" message accompanied by three asterisks. When you press ENTER or any other interrupt key, PDF returns to the previous panel.

MEMBER PARTS LIST

If you select the member parts list option from the batch selection panel, the panel shown in Figure 101 is displayed.

```

----- BATCH MEMBER PARTS LIST -----
OPTION ==>

  1 - Print member parts
  2 - Write member parts data set

ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>           ==>           ==>           ==>
  TYPE    ==>
  MEMBER  ==>           (Blank for member list, * for all members)

LANGUAGE  ==>           (Defaults to type value)

GROUPS FOR PRIMARY MEMBERS ==> (1, 2, 3, or 4)

SYSOUT CLASS ==>       (Defaults to 'A')

OUTPUT DATA SET:
  DATA SET NAME ==>   (Option 2 only)

```

Figure 101. Batch Member Parts List Panel

The difference between this panel and the foreground member parts listing panel is that the browse option is not allowed in batch. Otherwise, the function operates the same as the foreground version. The listing is 120 characters wide and uses ANSI printer controls.

LIST DATA SET

If you are using the same LIST dataset for multiple job steps, the DCB information may differ between the language processors and the linkage editor, causing an I/O error when attempting to read the LIST dataset. You should use a different LIST ID for each job step.

CHAPTER 10. COMMAND (OPTION 6)

The command option allows direct entry of a TSO command or command procedure (CLIST) under PDF. When you select this option, a panel (Figure 102) is displayed.

Note: PDF allows you to enter TSO commands and CLISTs in the command field of any panel. Option 6 provides two additional capabilities:

- You can enter a long command that continues on the following two lines.
- You can optionally enter Session Manager mode.

```
----- TSO COMMAND PROCESSOR -----  
  
ENTER TSO COMMAND OR CLIST BELOW:  
==>
```

Figure 102. TSO Command Entry Panel

On the command line, enter the command or CLIST to the right of the arrow, using standard TSO syntax. You may continue it on the next two

lines, if necessary, up to a maximum of 234 characters. You can also enter PDF commands, such as RETURN or END, in this field.

Notes:

1. You need not enter "TSO" before the command, as you do on other panels, unless the command to be executed exists in both ISPF and TSO and you want to execute the TSO command. If you use "TSO", your executed command is blanked out when the command panel is redisplayed.
2. If you enter the HELP command, it is interpreted as the ISPF HELP command. To obtain TSO HELP information, enter:

==> TSO HELP xxx

You can use H as an abbreviation for HELP in the TSO HELP command.

3. If you enter the PRINT command, it is interpreted as the ISPF PRINT command. To execute the TSO PRINT command, enter:

==> TSO PRINT xxx

Do not enter the following commands under PDF:

- LOGON, LOGOFF
- ISPF, PDF, and ISPSTART
- TEST
- Commands that are restricted by TSO or PDF
- Commands that require APF authorization (such as RACF commands)

Note: These commands are permissible if the TSO/E Release 2 Program Product is installed.

- Commands that invoke an APF-authorized program
- Calls to the ISPEXEC dialog service

You can enter a CLIST name on this panel, but the following restrictions apply:

- The CLIST cannot invoke the restricted commands shown in the preceding list.

Note: This does not apply to ISPEXEC, which may be invoked in a CLIST.

- CLIST attention exits produce unpredictable results when you press PA1.

Command (Option 6)

- CLIST error exits are not entered for ABENDs.
- TERMIN command procedure statements cause unpredictable results.

Note: A command issued via its synonym can cause unpredictable results.

If your installation has the Session Manager Program Product installed, the panel includes an entry line:

```
ENTER SESSION MANAGER MODE ==>          (YES or NO)
```

immediately below the line on which you enter your command or CLIST name.

If you specify YES, and if you are set up to use the Session Manager, any display output is displayed in the Session Manager TSOOUT stream. The PF key definitions are not transferred to the Session Manager from PDF. When the command terminates, the Session Manager prompts you to enter a null line to return to PDF control and redisplay the command panel. If you specify NO, terminal input/output occurs as though the Session Manager were not installed.

If you do not select Session Manager control, the terminal is operated in normal TSO fashion. Any communication with the command is in line-I/O mode. Upon completion of the command, three asterisks (***) are displayed. Press ENTER to redisplay the command panel in full screen mode.

To interrupt a TSO command or CLIST, press the PA1 key. The TSO command is terminated and the command panel redisplayed. If terminal input is inhibited, press the RESET key before pressing the PA1 key. If you are in Session Manager mode, enter a null line to return to ISPF full screen mode.

When the command 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.

For terminals with primary and alternate screen sizes, the user should be aware that ISPF does not check to make sure the same terminal settings are in effect upon completion of a command or CLIST. If the user invokes a CLIST or command that causes the screen settings to be changed, then the user is responsible to make sure the screen settings are saved and restored before control is returned to ISPF.

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 TSO TEST 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 103) 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 103. 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 logical screens (when you are using split screen mode), 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 for MVS Dialog Management Services.

DIALOG TEST VARIABLE USAGE

When you select the dialog test option, you are given a new function pool, a new shared variable pool, and the **ISRPROF** profile pool. 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.

Note: If you set a dialog variable in the shared pool from a dialog running under dialog test and then reinvoke it from the command line, you will not be able to retrieve the value of that variable.

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.

Note: ISPF does not support TSO global variables and you may encounter a severe dialog test error when testing a dialog that references a global variable.

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. If the LOCATE argument ends in an asterisk (*), a scan for the characters preceding the asterisk is done. 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 activity completes execution, you return to the invoke function entry panel. The old activity resumes 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 104).

- 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 (CLIST), or any TSO command, to be invoked as a dialog function. You may include command parameters.

LANG Enter APL to specify the use of the APL language. If this is your first request during the session, the command specified in the CMD keyword is invoked and an APL2 environment is established. If this is not your first APL request during this session, the string specified after the CMD keyword is passed to the APL2 workspace and executed.

To specify any language other than APL, leave this field blank.

- 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.

PARAM 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.

Dialog Test Functions (Option 7.1)

no, n This function is not a new application.

ID A 1- to 4-character ID for a new application. If you invoke a new application and leave the ID field blank, the default ID of ISR 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.

```
----- INVOKE DIALOG FUNCTION/SELECTION MENU -----
COMMAND ==>

INVOKE SELECTION MENU:
  PANEL ==> _                               OPT ==>

INVOKE COMMAND:
  CMD ==>

  LANG ==>                                 (APL or blank)

INVOKE PROGRAM:
  PGM ==>                                 PARM ==>

NEWAPPL ==> NO                             ID ==>

NEWPOOL ==> NO
```

Figure 104. Dialog Test - Invoke Function Entry Panel

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 written to the current function variable pool. When you select the panels option, you are shown a panel (Figure 105).

```

----- DISPLAY PANEL -----
COMMAND ==>

PANEL NAME      ==>

MESSAGE ID      ==>          (Optional)

CURSOR          ==>          (Optional)

CURSOR POSITION  ==>          (Optional)

```

Figure 105. Dialog Test - Display Panel Entry Panel

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).
CURSOR POSITION	An integer specifying the position in the field where the cursor is to be placed (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 or cursor position without identifying a panel name, the message is displayed on a separate panel (Figure 106). 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. 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, leave dialog test, or reset the function pool.

----- MESSAGE DISPLAY PANEL ----- SHORT MESSAGE
COMMAND ==> _

This panel is used to display your specified message.

Figure 106. 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 107) indicates all the current variables for the dialog being tested.

```
----- VARIABLE DISPLAY AND SET ----- ROW 1 of 72
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 107. Dialog Test - Variables Entry Panel

Each line of the display represents a variable and contains:

'''' Line command area.

VARIABLE Name of the variable: an alphanumeric 8-character name, with the first character alphabetic, or the special characters '\$', '#', or '@'. The alphabetic characters are translated

Dialog Test Variables (Option 7.3)

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.

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

Dialog Test Variables (Option 7.3)

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.

DBCS DATA

DBCS is a variable defined by the VDEFINE service or displayed through the field with FORMAT(DBCS) specified in the test environment. The DBCS value is displayed using the form '[DBDBDB]', where '[' and ']' represent the SO and SI characters, respectively.

If you enter a DBCS value in this format on the 'Dialog Test - Variables Entry Panel,' only the DBCS characters are stored.

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 108) 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 row

TABLE NAME      ==>          CURRENT ROW:

ROW IDENTIFICATION:
  BY ROW NUMBER ==> *      (*=Current row)

  BY VARIABLE     VALUE    (Search for row if row number blank)
  _____
  _____
  _____
  _____
  _____

DBCS COLUMN SPECIFICATION:
  _____  _____  _____  _____  _____  _____
  _____  _____  _____  _____  _____  _____
```

Figure 108. 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 row
	5 Display structure
	6 Display status

Dialog Test Tables (Option 7.4)

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	Overtyping 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 be used in the search, up to 58 characters. For an abbreviated search, enter the beginning characters followed by an asterisk (*). You can specify a DBCS value in the form '-[DBDBDB]', where '[' and ']' represent the SO and SI characters, respectively. For an abbreviated search, enter a two-byte asterisk (*) at the end of the DBCS value: for example, '-[DBDB**]', where '**' represents the two-byte asterisk character.
DBCS COLUMN SPECIFICATION	Enter the variable names of the values that are DBCS data. The value of the variable is displayed using the form '-[DBDBDB]'. If you enter a DBCS value in this format on the 'Dialog Test - Modify Table Row Entry Panel' or the 'Dialog Test - Add Table Row Entry Panel', only the DBCS characters are stored, regardless of the DBCS column specification.

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 109). In the figure, the variables constitute one table row.

```

DISPLAY ROW TABLE SAMTB1 ROW 3 ----- ROW 3 OF 21
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 variable N2 is also a save variable
SAVE1____ S Save variable 1 has this value
SAVE2____ S Show save variable 2 this way
N2_____ S Save variable N2 is also a name variable
***** BOTTOM OF DATA *****

```

Figure 109. Dialog Test - Display Table Row Entry Panel

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 The first 58 characters of the variable value

The display row option uses the END and LOCATE dialog test primary commands. These commands are described in the section entitled "Dialog Test Commands."

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 110) 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 110. 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 open table and identifying an existing row of the table.

When you select this option, a scrollable display (Figure 111) 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

T          Type of variable, nonmodifiable:

          K Key variable
    
```

```

MODIFY ROW TABLE SAMTB1 ROW 6 ----- ROW 6 OF 21
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 variable N2 is also a save variable
'''' SAVE1___ S Save variable 1 has this value
'''' SAVE2___ S Show save variable 2 this way
'''' N2_____ S Save variable N2 is also a name variable
'''' _____ S
'''' _____ S
'''' _____ S
'''' _____ S
'''' _____ S
'''' _____ S
'''' _____ S
    
```

```

***** BOTTOM OF DATA *****
    
```

Figure 111. Dialog Test - Modify Table Row Entry Panel

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.
- If the table was sorted using the TBSORT dialog service and a sort field is modified, the rows position in the table may change to preserve the search order.
- 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.

Dialog Test Tables (Option 7.4)

- 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 Row (Tables Option 4)

The add 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 112), 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 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.
- The position of the new row in the table depends on whether the table was previously sorted using the TBSORT dialog service. If the table was sorted, the new row is placed in sort order; if it has not been sorted, the new row is placed after the row you specified.

```
ADD ROW TABLE SAMTB1 ROW 7 ----- ROW 7 OF 18
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
'''' _____ S
'''' _____ S
'''' _____ S
'''' _____ S
***** BOTTOM OF DATA *****
```

Figure 112. Dialog Test - Add Row Entry Panel

Dialog Test Tables (Option 7.4)

- 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 was sorted using the TBSORT dialog service, the rows position in the table may change to preserve the search order.
- 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 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 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 113) 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 113. 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 a status panel (Figure 114). The panel provides 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 (date is in the national format)
TIME CREATED	Time the table was created
LAST DATE MODIFIED	Date the table was last modified (date is in the national format)
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 : NOT OPEN      DATE CREATED       : 83/06/15
TABLE AVAILABLE         : YES           TIME CREATED        : 10:15
                                                                    LAST DATE MODIFIED: 83/06/25
                                                                    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 114. Dialog Test - Table Status Display for a Closed 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.

Dialog Test Tables (Option 7.4)

If the table is open for your user id, you are shown a status panel (Figure 115). This panel provides 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 (date is in the national format)
TIME CREATED	Time the table was created
LAST DATE MODIFIED	Date the table was last modified (date is in the national format)
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.

----- STATUS FOR TABLE SAMTB1 -----

STATUS FOR THIS SCREEN	: OPEN	DATE CREATED	: 83/06/15
OPEN OPTION	: WRITE	TIME CREATED	: 10:15
TABLE ON DISK	: YES	LAST DATE MODIFIED:	83/06/25
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	: 4

Figure 115. Dialog Test - Table Status Display for an Open Table

LOG (OPTION 7.5)

The log option allows you to display and browse data recorded in the ISPF log. You can use all the browse commands (except BROWSE) 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 log file was preallocated with a disposition of OLD. The log file must be allocated with a disposition of MOD.
- The log file has been preallocated to SYSOUT.

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 and ISPLNK calls (except for the ISREDIT service) 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').

DIALOG SERVICES (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 116 shows the invoke dialog service panel.

```
----- INVOKE DIALOG SERVICE -----  
COMMAND ==>  
  
ENTER DIALOG SERVICE AND ITS PARAMETERS:  
==> _
```

Figure 116. Dialog Test - Invoke Dialog 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 117).

----- SPECIAL DISPLAY PANEL -----
COMMAND ==> _

This panel is used for two special "DISPLAY" conditions:

1. When DISPLAY is invoked without a panel name.
2. When TBDISPL is invoked without a panel name. All of the other parameters are ignored.

Figure 117. 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 118) 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 118: Dialog Test - Traces Selection Panel

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 120) 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.

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 121), 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

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 123 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 123. 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:

Dialog Test – Encountering a Breakpoint

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 123, 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 three 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'). ISPEXEC calls from a program are the same as ISPEXEC calls from a command except that ISPEXEC is not displayed.
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 redisplayes 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. LIBRARY MANAGEMENT UTILITIES (OPTION 8)

When you select option 8 from the ISPF/PDF primary option menu, the panel shown in Figure 124 is displayed. The options on this panel allow you to:

- Specify library attributes
- Create or update lists of authorized users and distribution tables for a project
- Activate or deactivate the controls for a project
- Browse or print activity and library control information

In order to use the library management utilities, you must have pre-allocated the control file using the ddname ISRCFIL.

If you create the controls for a library for a new project, you automatically become the project administrator for that project. The project administrator is the only person who can update the library controls and execute options 2, 3, and 4 for that project. The project administrator of a project can change the project administrator id of the project. Where there are several project administrators for a given project, only one administrator can update the project controls. The updating of project controls is restricted to one logical screen.

For complete information on the use of the library management utilities, see ISPF/PDF for MVS Library Management.

```

----- LIBRARY MANAGEMENT UTILITIES -----
OPTION ==> _

1 CONTROLS      - Specify library attributes:
                  Create, update, or delete promotion hierarchies,
                  Authorize access to controls or members in hierarchies,
                  Specify action taken when promoting members.
2 USER SET     - Create, update, or delete list of users authorized to
                  perform library functions.
3 DISTRIBUTION - Create, update, or delete a distribution table to relate
                  different promotion hierarchies.
4 ACTIVATE     - Activate or deactivate promotion hierarchy controls.
5 REVIEW       - Browse or print activity and library controls
                  information.
T TUTORIAL     - General information about the library management
                  utility facility.

```

Figure 124. Library Management Utility Panel

CHAPTER 13. 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 125). 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 125 (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 125 (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 brief scenario is an example of PDF terminal operations. It is intended for new users as a quick introduction to PDF. It may also serve as a demonstration that PDF has been properly installed and is operational.

This scenario accesses data set SYS1.SAMPLIB, which should contain the following four members. These members are distributed with PDF.

ISRASM	- Sample Assembler Source
ISRCOBOL	- Sample COBOL Source
ISRFORT	- Sample FORTRAN Source
ISRPLI	- Sample PL/I Source

During the scenario, member ISRASM is copied from SYS1.SAMPLIB to a user data set. The other three members are not used during the scenario.

The scenario also assumes the existence of the TSO Assembler Prompter, program number 5734-CP2. If the Assembler Prompter is not available at your installation, skip steps 34 through 40.

ACTION	RESULT
1. LOGON TO TSO	1. "READY"
2. KEY "PDF"	2. ISPF/PDF PRIMARY OPTION MENU
3. SELECT UTILITIES (OPTION 3)	3. UTILITY SELECTION PANEL
4. SELECT DATA SET UTILITY (OPTION 2)	4. DATA SET UTILITY PANEL
5. SELECT OPTION A (ALLOCATE) AND SPECIFY ISPF LIBRARY: PROJECT : SAMPLE GROUP : USERID (YOUR TSO USERID) TYPE : ASM	5. ALLOCATE NEW DATA SET PANEL
6. ENTER THE FOLLOWING: VOLUME : (BLANK) GENERIC UNIT: (BLANK) UNITS : TRKS PRIMARY : 2 SECONDARY : 1 DIR BLKS : 1 FORMAT : FB LENGTH : 80 BLK SIZE : 3120	6. DATA SET SAMPLE.USERID.ASM IS ALLOCATED ON SCRATCH VOLUME. DATA SET UTILITY PANEL IS DISPLAYED.
7. SELECT OPTION A AGAIN AND SPECIFY ISPF LIBRARY: PROJECT : SAMPLE GROUP : USERID (YOUR TSO USERID) TYPE : OBJ	7. ALLOCATE NEW DATA SET PANEL
8. LEAVE EVERYTHING THE SAME. JUST PRESS ENTER.	8. DATA SET SAMPLE.USERID.OBJ IS ALLOCATED. DATA SET 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 SPECIFY "OTHER" DATA SET NAME: 'SYS1.SAMPLIB(ISRASM)'	12. SECOND MOVE/COPY PANEL

ACTION	RESULT
13. SPECIFY "TO" ISPF LIBRARY: PROJECT : SAMPLE GROUP : USERID TYPE : ASM MEMBER : (BLANK)	13. MEMBER ISRASM IS COPIED FROM DATA SET SYS1.SAMPLIB TO SAMPLE.USERID.ASM. 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. SPECIFY ISPF LIBRARY: PROJECT : SAMPLE GROUP : USERID TYPE : ASM MEMBER : (LEAVE BLANK)	16. BROWSE MEMBER LIST OF DATA SET SAMPLE.USERID.ASM
17. SELECT MEMBER ISRASM (ENTER "S" AHEAD OF MEMBER NAME)	17. BROWSE TEXT DISPLAY OF SAMPLE.USERID.ASM(ISRASM)
18. SCROLL FORWARD ONE PAGE (PF8)	18. SECOND PAGE OF ISRASM
19. SCROLL BACKWARD ONE PAGE (PF7)	19. FIRST PAGE OF ISRASM 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 GROUP : USERID TYPE : ASM MEMBER : ISRASM	26. EDIT DATA DISPLAY OF DATA SET SAMPLE.USERID.ASM(ISRASM)

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 ISRASM IS SAVED IN DATA SET SAMPLE.USERID.ASM AND A LISTING OF THE MEMBER IS PLACED IN THE ISPF LIST DATA SET. 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. SPECIFY ASSEMBLER INPUT: PROJECT : SAMPLE GROUP : USERID TYPE : ASM MEMBER : ISRASM LIST ID : ASMTEST1 ASSEMBLER PARAMETERS: LIST,TEST,RENT	36. ASSEMBLER PROMPTER IS INVOKED. TERMINAL OUTPUT IS WRITTEN STARTING ON THE LOWER PART OF THE SCREEN. WHEN "***" 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 FILL IN JOB CARD INFORMATION AS REQUIRED BY YOUR INSTALLATION.	42. BATCH ASSEMBLY PANEL
43. SPECIFY ASSEMBLER INPUT: PROJECT : SAMPLE GROUP : USERID TYPE : ASM MEMBER : ISRASM LISTID : ASMTEST2 ASSEMBLER PARAMETERS: OBJ,NODECK,LIST,TERM	43. JCL FOR ASSEMBLING ISRASM IS GENERATED. THEN THE BATCH SELECTION PANEL IS DISPLAYED.
44. PRESS PF3	44. THE JOB IS SUBMITTED. JOBNAME IS DISPLAYED AT LOWER PART OF SCREEN. WHEN "***" APPEARS PRESS ENTER. THE ISPF/PDF PRIMARY OPTION MENU IS DISPLAYED.
45. SELECT TSO COMMAND (OPTION 6)	45. TSO COMMAND ENTRY PANEL
46. ENTER THE FOLLOWING COMMAND: ==> STATUS	46. STATUS OF YOUR JOB IS DISPLAYED. WHEN "***" APPEARS PRESS ENTER. TSO COMMAND PANEL IS RE-DISPLAYED WITH PREVIOUS COMMAND PRE-ENTERED.

ACTION	RESULT
47. PRESS PF3	47. ISPF/PDF PRIMARY OPTION MENU
48. SELECT TUTORIAL (OPTION T)	48. START OF ISPF/PDF TUTORIAL IS DISPLAYED. FOLLOW THE DIRECTIONS TO LEARN MORE ABOUT ISPF/PDF. WHEN YOU ARE DONE, HIT PF3 TO RETURN TO THE ISPF/PDF PRIMARY OPTION MENU.
<p>IF YOU HAVE FOLLOWED THIS SCENARIO, YOU SHOULD HAVE AN ASSEMBLER LISTING IN DATA SET USERID.ASMTEST1.LIST. ALSO, WHEN THE JOB THAT YOU SUBMITTED HAS FINISHED, YOU WILL HAVE ANOTHER LISTING IN DATA SET USERID.ASMTEST2.LIST. TRY BROWSE (OPTION 1) TO REVIEW THE LISTINGS, AND TRY EXPERIMENTING WITH OTHER ISPF/PDF OPTIONS. WHEN YOU ARE DONE, RETURN TO THE ISPF/PDF PRIMARY OPTION MENU.</p>	
49. (AT PRIMARY OPTION MENU) PRESS PF3	49. TERMINATION PANEL
50. SELECT OPTION "PD" FOR BOTH THE LOG AND LIST DATA SETS AND FILL IN JOB CARD INFORMATION AS REQUIRED BY YOUR INSTALLATION. THEN PRESS ENTER.	50. A JOB IS SUBMITTED TO PRINT AND DELETE THE DATA SETS. JOBNAME IS DISPLAYED AT LOWER PART OF SCREEN. WHEN "***" APPEARS PRESS ENTER. SCREEN IS CLEARED AND TSO "READY" MESSAGE IS DISPLAYED.
51. YOU ARE NOW OUT OF ISPF. TO EXIT FROM TSO, ENTER "LOGOFF".	51. END OF USAGE SCENARIO

APPENDIX B. PDF LISTING FORMATS

To assist in the maintenance of hardcopy listings, the PDF editor (under user option) automatically provides a source listing of each segment upon completion of editing if autolist mode is on. Source listings may also be obtained via the Print option of the PDF library utility and the move/copy utility. PDF provides index listings, upon user request, via the Index option of the library utility, and catalog and VTOC listings via the Print options of the catalog management and list VTOC utilities.

A sample source listing is shown in Figure 126.

Information at the top of the page includes project, group, type, and member name, current version and modification level, user ID, date and time that the listing was produced, and page number. The member name is also repeated at the right for ease of filing. 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. 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 (**).

A sample index listing for a source library is shown in Figure 127. 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 data set, including current space allocation and utilization. Following this, the member name and statistics are printed for each member in the data set, arranged in alphabetical order. (For sequential data sets, the index listing contains just the general information.)

An index listing for an OBJ library is similar except that no statistics are maintained. A sample index listing for a LOAD library is shown in Figure 128. Here, the module attributes are printed to the right of each member name.

A sample ISPF log listing is shown in Figure 129. The log contains a message for each significant user action, such as saving edited data, moving members from one data set to another, or submitting a background job.

A sample data set list listing is shown in Figure 130. The data set list contains the volume serial on which each data set resides and other data set characteristics, when available.

PROJECT: SPFDEMO
GROUP: MYLIB
TYPE: PLI

MEMBER: COINS
LEVEL: 01.04
USERID: JOSLIN

DATE: 83/05/18
TIME: 17:22
PAGE: 01 OF 01 START

COINS

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 126. Sample Source Listing

PROJECT: SPFDEMO
 GROUP: MYLIB
 TYPE: PLI

DATE: 84/01/06
 TIME: 14:10
 PAGE: 001

GENERAL DATA:
 VOLUME SERIAL: D00163
 DEVICE TYPE: 3330
 ORGANIZATION: PO
 CREATION DATE: 80/11/07
 EXPIRATION DATE: **NONE**

GENERAL DATA:
 RECORD FORMAT: FB
 RECORD LENGTH: 80
 BLOCK SIZE: 1,600
 1ST EXTENT SIZE: 20
 SECONDARY QUAN: 0

CURRENT ALLOCATION:
 20 TRACKS
 1 EXTENT
 10 DIRECTORY BLOCKS

CURRENT UTILIZATION:
 5 TRACKS
 1 EXTENT
 2 DIRECTORY BLOCKS
 10 MEMBERS

MEMBER NAME	X VERS.MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED	CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
ACCOUNT	01.00	84/01/09	84/01/09 17:07	21	21	0	HUNTER
ACCT1	01.01	84/01/09	84/04/23 14:52	99	193	0	BECKETT
ACCT2	01.00	84/01/09	84/01/09 17:07	20	20	0	BECKETT
COINS	01.04	84/04/24	84/04/28 16:20	19	19	4	JOSLIN
COMPX	01.00	84/01/09	84/01/09 17:08	44	44	0	JOSLIN
COMPY	01.01	84/01/14	84/01/16 12:30	13	13	1	JOSLIN
DCLS	01.00	84/04/23	84/04/23 15:14	20	20	0	LEESB
LISTOUT	01.02	84/04/23	84/05/06 15:00	17	13	6	LEESB
MAIN	01.00	84/01/09	84/01/09 17:08	4	4	0	KIRK
TESTDIR	01.02	84/04/23	84/05/06 17:04	30	43	10	LAIDLAW
MAXIMUMS:	01.04	84/04/24	84/05/06 17:04	99	193	10	
TOTALS:				287	390	21	

END OF MEMBER LIST

Figure 127. Sample Index Listing - Source Library

PROJECT: SPFDEMO
 GROUP: MYLIB
 TYPE: LOAD

DATE: 84/01/08
 TIME: 09:35
 PAGE: 001

GENERAL DATA:	GENERAL DATA:	CURRENT ALLOCATION:	CURRENT UTILIZATION:
VOLUME SERIAL: D00172	RECORD FORMAT: U	88 BLOCKS	86 BLOCKS
DEVICE TYPE: 3330	RECORD LENGTH: 0	12 EXTENTS	12 EXTENTS
ORGANIZATION: PO	BLOCK SIZE: 3,072	4 DIRECTORY BLOCKS	2 DIRECTORY BLOCKS
CREATION DATE: 83/11/05	1ST EXTENT SIZE: 44		8 MEMBERS
EXPIRATION DATE: **NONE**	SECONDARY QUAN: 4		

MEMBER NAME	ALIAS OF	SSI	LOAD MODULE (HEX)	SIZE (DEC)	ENTRY POINT (HEX)	TTR (HEX)	AUTH CODE	MODULE	----- ATTRIBUTES -----
SPFEDIT			000148	328	000000	000006	00	FO	RF RN RU
SPFMAIN			004F98	20,376	000000	00000D	00	FO	RF RN RU
SPFPMD			0016E8	5,864	000000	000204	00	FO	RF RN RU
SPFSUBS			024698	149,144	000000	000211	00	FO	RF RN RU
SPFTBLS			002470	9,328	000000	00110B	00	FO	
SPFUDA			0046D8	18,136	000000	00120A	00	FO	RF RN RU
SPFURS			000FC0	4,032	000000	00140A	00	FO	RF RN RU
SPF3277			000548	1,352	000000	001504	00	FO	RF RN RU
TOTALS:			00032E80	208,560					

END OF MEMBER LIST

 MODULE ATTRIBUTE CODES:
 FO PROCESS ONLY BY F LEVEL LINKAGE EDITOR
 NX NOT EXECUTABLE
 OL ONLY LOADABLE
 OV IN OVERLAY STRUCTURE
 RF REFRESHABLE
 RN REENTERABLE
 RU REUSABLE
 SC SCATTER FORMAT
 TS MODULE TO BE TESTED

Figure 128. Sample Index Listing - Load Library

```

TIME          *** SPF TRANSACTION LOG ***                USERID: WALTERS   DATE: 83/12/22   PAGE: 1

17:11  START OF ISPF SESSION LOG #64 -----
17:12  UTILITY - ALLOCATE - ISPFDEMO.NEWLIB.PLI ALLOCATED ON VOLUME WORK97
17:13  UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI (ACCOUNT) COPIED
17:13                                     TO ISPFDEMO.NEWLIB.PLI (ACCTXX)
17:13  UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI (ACCT1) COPIED
17:13                                     TO ISPFDEMO.NEWLIB.PLI (ACCTYY)
17:13  UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI (ACCT2) COPIED
17:13                                     TO ISPFDEMO.NEWLIB.PLI (ACCTZZ)
17:13  UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI (COINS) COPIED
17:13                                     TO ISPFDEMO.NEWLIB.PLI (COINS)
17:14  EDIT      - SAVE      - ISPFDEMO.NEWLIB.PLI (COINS) - MEMBER SAVED
17:15  EDIT      - CREATE    - ISPFDEMO.NEWLIB.PLI (ACCTNEW) - MEMBER CREATED
17:15  EDIT      - SAVE      - ISPFDEMO.NEWLIB.PLI (ACCTXX) - MEMBER SAVED
17:15  EDIT      - SAVE      - ISPFDEMO.NEWLIB.PLI (ACCTYY) - MEMBER SAVED
17:16  EDIT      - SAVE      - ISPFDEMO.NEWLIB.PLI (ACCTZZ) - MEMBER SAVED
17:17  UTILITY - DELETE    - ISPFDEMO.NEWLIB.PLI (ACCTZZ) DELETED
17:17  TSO       - COMMAND   - LISTC
17:17  UTILITY - ALLOCATE - ISPFDEMO.NEWLIB.OBJ ALLOCATED ON VOLUME WORK98
17:19  JOB      - WALT8     - PLI OPT COMPILE -- TO ISPFDEMO.NEWLIB.OBJ (COINS)
17:19  JOB      - WALT8     - SUBMITTED
17:20  END OF ISPF SESSION LOG #64 -----

```

Figure 129. Sample Log Listing

LISTING OF DATA SETS BEGINNING WITH ISPFPROJ.MASTER

DATE: 84/02/15
 TIME: 13:03
 PAGE: 1

DATA SET NAME	VOLUME	ORG	RECFM	LRECL	BLKSZ	TRKS	%USED	XT	CREATED
ISPFPROJ.MASTER.ASM	IS5038	PO	FB	80	800	1	100	1	83/04/30
ISPFPROJ.MASTER.CLIST	IS5038	PO	VB	255	3120	40	100	4	83/04/30
ISPFPROJ.MASTER.DATASETS	HSMDDMP								
ISPFPROJ.MASTER.DEVGUIDE	IS5038	PS	FBM	81	12960	17	100	4	83/05/15
ISPFPROJ.MASTER.EDITLOG	HSMDDMP								
ISPFPROJ.MASTER.EDITLOGO	HSMDDMP								
ISPFPROJ.MASTER.HELP	IS5038	PO	FB	80	3120	4	75	4	84/03/29
ISPFPROJ.MASTER.JCL	IS5038	PO	FB	80	12960	10	100	1	83/03/25
ISPFPROJ.MASTER.LEL	IS5038	PO	FB	80	3120	2	100	1	83/04/09
ISPFPROJ.MASTER.LISTS	IS5038	PO	FB	80	12960	48	60	4	83/04/07
ISPFPROJ.MASTER.LOAD	GB5311	PO	U	256	3072	37	100	1	83/05/08
ISPFPROJ.MASTER.MSGS	IS4002	PO	FB	80	3120	4	100	2	83/03/30
ISPFPROJ.MASTER.OBJ	IS5038	PO	FB	80	3120	11	100	2	83/04/09
ISPFPROJ.MASTER.PANELS	IS5038	PO	FB	80	3120	33	87	4	83/05/29
ISPFPROJ.MASTER.PLS	IS5038	PO	FB	80	12960	63	100	4	83/04/03
ISPFPROJ.MASTER.PROMLOG	TSOPK5	PS	FB	100	4000	6	100	6	83/11/22
ISPFPROJ.MASTER.PROMLOG0	TSOPK4	PS	FB	100	4000	8	100	1	83/08/07
ISPFPROJ.MASTER.PROMLOG1	TSOPK4	PS	FB	100	4000	9	100	1	83/08/29
ISPFPROJ.MASTER.PROMLOG2	TSOPK4	PS	FB	100	4000	10	100	1	83/10/04
ISPFPROJ.MASTER.PROMLOG3	TSOPK7	PS	FB	100	4000	16	100	1	83/11/03
ISPFPROJ.MASTER.PROMLOG4	TSOPK6	PS	FB	100	4000	17	100	1	83/12/21
ISPFPROJ.MASTER.PROMLOG5	TSOPK7	PS	FB	100	4000	6	100	1	84/01/18
ISPFPROJ.MASTER.PROMLOG7	IS5282	PS	FB	100	4000	15	86	1	83/06/10
ISPFPROJ.MASTER.PROMLOG8	MIGRAT								
ISPFPROJ.MASTER.PROMLOG9	TSOPK4	PS	FB	100	4000	8	100	1	83/07/14
ISPFPROJ.MASTER.PTMLOG	IS5038	PS	FB	80	3120	48	10	10	83/07/28
ISPFPROJ.MASTER.PTMLOG1	HSMDDMP								
ISPFPROJ.MASTER.PTMLOG2	IS5038	PS	FB	80	3120	57	100	12	84/02/25

Figure 130. Sample Data Set List Listing

APPENDIX C. CHARACTER TRANSLATIONS FOR APL AND TEXT KEYBOARDS

You can use APL keyboards for all models of 3270 terminals, and text keyboards for 3278 and 3279 terminals with PDF. 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 131 and Figure 132.

	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	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>	<u>I</u>	¢	.	<	(+		4F
50	&	<u>J</u>	<u>K</u>	<u>L</u>	<u>M</u>	<u>N</u>	<u>O</u>	<u>P</u>	<u>Q</u>	<u>R</u>	!	\$	*)	;	~	5F
60	-	/	<u>S</u>	<u>T</u>	<u>U</u>	<u>V</u>	<u>W</u>	<u>X</u>	<u>Y</u>	<u>Z</u>		,	%	_	>	?	6F
70		^	..	⊖	⊗	⊘		v	'	:	#	@	'	=	"		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	α	ε	ι	ρ	ω		x	\	÷		∇	Δ	τ]	≠		BF
C0	{	A	B	C	D	E	F	G	H	I	∞	∞	∅	∅	∅	∅	CF
D0	}	J	K	L	M	N	O	P	Q	R	I	!	∇	∇	∇	∇	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 131. 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	o								'	:	#	@	'	=	"	7F
80		a	b	c	d	e	f	g	h	i	↑	{	≤	(+	+	8F
90	□	j	k	l	m	n	o	p	q	r		}	π)	±	■	9F
A0	-	~	s	t	u	v	w	x	y	z	●	L	Γ	[≥	•	AF
B0	°	1	2	3	4	5	6	7	8	9	∇	┘	┘]	≠	-	BF
C0	{	A	B	C	D	E	F	G	H	I	Δ	⊥	τ	v	→	→	CF
D0	}	J	K	L	M	N	O	P	Q	R	⊞	Δ	§	¶	←	→	DF
E0	\	\	S	T	U	V	W	X	Y	Z	~v	F	+	▲	▼	∧	EF
F0	0	1	2	3	4	5	6	7	8	9		L	φ	φ	Γ		FF


 National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Figure 132. Internal Character Representations for Text Keyboards

APPENDIX D. 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 printed on a 328x printer or other device that does not have a suppress-space carriage control.

TSO Allows you to enter a TSO command or CLIST 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. **PANELID** is set to OFF at the beginning of a PDF session.

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

PRINTLHI 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 printed on a 328x printer or other device that does not have a suppress-space carriage control.

SPLITV On 3290 terminals, causes the screen to be split vertically.

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.

- BROWSE** Causes recursive browsing of another member of the data set, or another data set.
- COLS** Causes display of a line that identifies column positions.
- DISPLAY** Allows the display of carriage control characters and nondisplayable characters.
- 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 input 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.
- AUTOSAVE** Causes data to be saved to disk when the END command is entered. AUTOSAVE OFF PROMPT causes the user to be prompted whether to save the data or cancel. AUTOSAVE OFF NOPROMPT causes no saving of data when the END command is entered.
- BOUNDS** Sets the left and right boundaries and saves them in the edit profile.
- BUILTIN** Allows execution of a PDF edit command even when an edit macro with the same name exists.
- 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, or with a line label. The other source can be a member of any partitioned data set, or a sequential data set.
- CREATE** Causes a line or block of lines to be stored in any partitioned data set 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, or with a pair of line numbers or line labels.
DEFINE	Identifies a name as an macro, either program or CLIST, assigns an alias to the name of a built-in command or macro, or disables the definition of a name as a macro.
DELETE	Removes one or more lines from the data being edited.
EDIT	Causes the editing of another member of the same data set, or of another data set, without leaving the current edit session.
EXCLUDE	Causes lines to be excluded on which a specified character string is found. The cursor is repositioned to the beginning of the first occurrence, if it is found.
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.
IMACRO	Causes the name of the initial macro to be issued at the beginning of an edit session to be saved in the edit profile.
LEVEL	Sets the modification level of data being edited.
LOCATE	Causes direct scrolling to the specified line number or to a particular type of line.
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, or with a label.
MOVE	Same as COPY, except that the other source is deleted after the copy operation.
NOTE	Causes tutorial notes to be included when the MODEL command is used.
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.
PACK	Causes data that is saved to be stored in packed format.

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. Can also define a new profile, switch to a different profile, or lock or unlock the current profile.

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 an entire sequential data set to be replaced.

RESET Terminates the display of all message lines, note lines, and other special lines, resets line numbers to normal after CHANGE commands or CHANGE errors, blanks out pending line commands, and redisplay all excluded lines.

RMACRO Causes the name of the recovery macro, issued after edit recovery has processed a file, to be saved in the edit profile.

SAVE Causes the data to be stored back into the edit data set and editing to continue.

SORT Orders data within the file that you are editing.

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 submitted to the job stream 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.

VERSION Sets the version number of data being edited.

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.
LC (lowercase)	Translates uppercase characters to lowercase characters on specified lines.
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.
- UC (uppercase)** Translates lowercase characters to uppercase characters on specified lines.
- 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 to indicate the beginning and end of a block of lines.

CC - Block copy	UCC - Block conversion
DD - Block delete	to uppercase
LCC - Block conversion	XX - Block exclude
to lowercase	<< - Block "data" shift left
MM - Block move	>> - Block "data" shift right
OO - Block overlay	((- Block "column" shift left
RR - Block repeat)) - 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

LCn - Convert uppercase alphabetic characters to lowercase for n lines, starting at this line

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
 UCn - Convert lowercase alphabetic characters to uppercase for
 n lines, starting at this line
 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.
- ! or !n** 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 E. 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

BROWSE [member]

COLS

DISPLAY [char] [CC]

DISPL [NOCC]

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]

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

BOUNDS [left-col right-col]
BOUND
BNDS
BND

BUILTIN command

CANCEL
CAN

CAPS [ON]
 [OFF]

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

COPY [member-name] {BEFORE label}
 {AFTER label}

CREATE [member-name] [line-range]
CRE

DEFINE name {CMD MACRO }
DEF {PGM MACRO }
 {ALIAS name2}
 {NOP }
 {RESET }

DELETE ALL range
DEL

DELETE {ALL X } [range]
DEL {ALL NX}

EDIT [member]

EXCLUDE string-1 [range] [NEXT] [CHARS] [X][col-1 [col-2]]
 EXCLUDED [ALL] [PREFIX] [NX]
 EXC [FIRST] [SUFFIX]
 EX [LAST] [WORD]
 X [PREV]

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

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

IMACRO [name]
 [NONE]

LEVEL number
 LEV

LOCATE {line-number}
 LOC {label }
 L

LOCATE [NEXT] [LABEL] [range]
 LOC [PREV] [COMMAND]
 L [FIRST] [ERROR]
 [LAST] [CHANGE]
 [SPECIAL]
 [EXCLUDED]

MODEL [CLASS [class-name]]
 MOD

MODEL [model-name [qualifier...]] [BEFORE label] [NOTE]
 MOD [AFTER label] [NONOTE]

MOVE [member-name] [{BEFORE label}]
 [{AFTER label}]

NOTES [ON]
 NOTE [OFF]

NULLS [ON] [ALL]
 NULL [OFF]
 NUL

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

PACK [ON]
 [OFF]

PROFILE [name] [number]
PROF
PRO
PR

PROFILE [LOCK]
PROF [UNLOCK]
PRO
PR

RECOVERY [ON]
RECOVER [OFF]
RECOVRY
RECVRY
RECOV
RECVR
REC

RENUM [STD] [COBOL] [DISPLAY]
REN

REPLACE [member-name] [line-range]
REPL
REP

RESET [LABEL] [COMMAND] [ERROR] [CHANGE] [SPECIAL] [EXCLUDED] [range]
RES [LAB] [COM] [ERR] [CHG] [SPE] [EX]
[C] [X]

RMACRO [name]
[NONE]

SAVE

SORT [range] [X] [sort-field1 ... sort-field5]
[NX]

STATS [ON]
[OFF]

SUBMIT [range]
SUB

TABS [ON] [tab-char] [ALL]
TAB [OFF]

UNNUMBER
UNNUMB
UNNUM
UNN

VERSION number
VERS
VER

FIND/CHANGE STRINGS (BROWSE AND EDIT)

Character String Format:

Simple string:	cccccc	(no embedded blanks or commas)
Delimited string:	'cccc'	or "cccc"
Hex string:	X'hhhh'	or 'hhhh'X
Text string:	T'cccc'	or 'cccc'T
Picture string:	P'ssss'	or 'sss's'P
Character string:	C'cccc'	or 'cccc'C
Previous string:	*	(single asterisk)

Picture Strings - Special Characters for String-1:

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

Picture Strings - Special Characters for String-2:

P'='	-	equal to corresponding character in string-1
P'>'	-	convert corresponding character in string-1 to uppercase
P'<'	-	convert corresponding character in string-1 to lowercase

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)

LC, LCC, LCn (lowercase)
 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)
 UC, UCC, UCn (uppercase)
 X, Xn, XX (exclude)
 <, <n, <<, <<n (data left)
 >, >n, >>, >>n (data right)
 (, (n, ((, ((n (cols left)
),)n,))),))n (cols right)

The following commands and aliases are provided for compatibility with prior releases of ISPF.

<u>Command</u>	<u>Equivalent</u>
ASIS	CAPS OFF
NONULLS	NULLS OFF
NO NULLS	
NONULL	
NONUL	
NONUMBER	NUM OFF
NONUMBR	
NONUM	
NOTABS	TABS OFF
NOTAB	
PRT	AUTOLIST

Note: PRT is an abbreviation for what was the EDIT PRINT command. PRINT is now a dialog manager command which causes the screen image to be printed.

APPENDIX F. 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

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

Table Functions (General)

TBCLOSE		Table close	ISREMG5
TBCREATE		Table create	ISREMG1
TBEND		Table end	ISREMG6
TBERASE		Table erase	ISREMG7
TBOPEN		Table open	ISREMG2
TBQUERY		Table query	ISREMG3
TBSAVE		Table save	ISREMG4
TBSTATS		Table statistics	ISREMG8

Table Functions (Row)

TBADD		Table row add	ISREMC1
TBBOTTOM		Table row pointer to bottom	ISREMCRA
TBDELETE		Table delete	ISREMC2
TBEXIST		Table exist	ISREMC6
TBGET		Table get	ISREMC3
TBMOD		Table modify	ISREMC5
TBPUT		Table put	ISREMC4
TBSARG		Table search argument	ISREMC7
TBSCAN		Table scan	ISREMC8
TBSKIP		Table skip	ISREMCRA
TBSORT		Table sort	ISREMCRD
TBTOP		Table top	ISREMC9
TBVCLEAR		Table 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	ISREMGD1
PQUERY		Obtain panel information	ISREMGD4
SETMSG		Set message display	ISREMGD3
TBDISPL		Table display information	ISREMGD2
Library Access			
LMCLOSE		Close a data set	ISREMGCL1
LMERASE		Erase a data set or library	ISREMGCL2
LMFREE		Release a data set	ISREMGCL3
LMGET		Read a record	ISREMGCL4
LMINIT		Establish a data id	ISREMGCL5
LMMADD		Add a member	ISREMGCL6
LMMDEL		Delete a member	ISREMGCL7
LMMFIND		Find a member	ISREMGCL8
LMMLIST		Create a member list	ISREMGCL9
LMMREN		Rename a member	ISREMGCLA
LMMREP		Replace a member	ISREMGCLB
LMOPEN		Open a data set	ISREMGCLC
LMPROM		Promote a member	ISREMGCLD
LMPUT		Write a record	ISREMGCLE
LMQUERY		Provide data set information	ISREMGCLF
LMRENAME		Rename a library	ISREMGCLG
Miscellaneous			
BROWSE		Browse service	ISREMGCM3
CONTROL		Control service	ISREMGCM2
EDIT		Edit service	ISREMGCM4
GETMSG		Get Message service	ISREMGCM6
LOG		Write message or log file	ISREMGCM5
SELECT		Select service	ISREMGCM1

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

ISPF SERVICES IN COBOL PROGRAMS

Working Storage

WORKSTOR		Working storage definition	ISREMBW1
----------	--	----------------------------	----------

Graphics

GRERROR		Graphics error block service	ISREMSB1
GRINIT		Graphics initialization	ISREMSB2
GRTERM		Graphics termination service	ISREMSB3

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
TBSTATS		Table statistics	ISREMBG8

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
TBSKIP		Table skip	ISREMBR8
TBSORT		Table sort	ISREMBRD
TBTOP		Table top	ISREMBR9
TBVCLEAR		Table 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

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

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
PQUERY		Obtain panel information	ISREMBD4
SETMSG		Set message display	ISREMBD3
TBDISPL		Table display information	ISREMBD2

Library Access

LMCLOSE		Close a data set	ISREMBL1
LMERASE		Erase a data set or library	ISREMBL2
LMFREE		Release a data set	ISREMBL3
LMGET		Read a record	ISREMBL4
LMINIT		Establish a data id	ISREMBL5
LMMADD		Add a member	ISREMBL6
LMMDEL		Delete a member	ISREMBL7
LMMFIND		Find a member	ISREMBL8
LMMLIST		Create a member list	ISREMBL9
LMMREN		Rename a member	ISREMBLA
LMMREP		Replace a member	ISREMBLB
LMOPEN		Open a data set	ISREMBLC
LMPROM		Promote a member	ISREMBLD
LMPUT		Write a record	ISREMBLE
LMQUERY		Provide data set information	ISREMBLF
LMRENAME		Rename a library	ISREMBLG

Miscellaneous

BROWSE	MVS	Browse service (MVS)	ISREMCB1
BROWSE	VM	Browse service (VM)	ISREMCB2
BROWSE	VSE	Browse service (VSE)	ISREMCB3
CONTROL		Control service	ISREMBM2
EDIT	MVS	Edit service (MVS)	ISREMCE1
EDIT	VM	Edit service (VM)	ISREMCE2
EDIT	VSE	Edit service (VSE)	ISREMCE3
GETMSG		Get message service	ISREMBM6
LOG		Write message or log file	ISREMBM5
SELECT		Select service	ISREMBM1

ISPF SERVICES IN EXECs

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

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
TBSTATS		Table statistics	ISREMEG8

Table Functions (Row)

TBADD		Table row add	ISREMER1
TBBOTTOM		Table row pointer to bottom	ISREMER4
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	ISREMER4
TBSORT		Table sort	ISREMERD
TBTOP		Table top	ISREMER9
TBVCLEAR		Table variable clear	ISREMER8

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	IREMMEF1

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
GETMSG		Get message service	ISREMEM6
LOG		Write message or log file	ISREMEM5
SELECT		Select service	ISREMEM1

ISPF SERVICES IN FORTRAN PROGRAMS

Graphics

GRERROR		Graphics error block service	ISREMF31
GRINIT		Graphics initialization	ISREMF32
GRTERM		Graphics termination service	ISREMF33

Table Functions (General)

TBCLOSE		Table close	ISREMF35
TBCREATE		Table create	ISREMF31
TBEND		Table end	ISREMF36
TBERASE		Table erase	ISREMF37
TBOPEN		Table open	ISREMF32
TBQUERY		Table query	ISREMF33
TBSAVE		Table save	ISREMF34
TBSTATS		Table statistics	ISREMF38

Table Functions (Row)

TBADD		Table row add	ISREMF31
TBBOTTOM		Table row pointer to bottom	ISREMF3A
TBDELETE		Table delete	ISREMF32
TBEXIST		Table exist	ISREMF36
TBGET		Table get	ISREMF33
TBMOD		Table modify	ISREMF35
TBPUT		Table put	ISREMF34
TBSARG		Table search argument	ISREMF37
TBSCAN		Table scan	ISREMF38
TBSKIP		Table skip	ISREMF3B
TBSORT		Table sort	ISREMF3D
TBTOP		Table top	ISREMF39
TBVCLEAR		Table variable clear	ISREMF3C

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

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

File Tailoring

FTCLOSE		End file tailoring	ISREFFF3
FTERASE		File tailor erase	ISREFFF4
FTINCL		File tailor include skeleton	ISREFFF2
FTOPEN		File tailor open	IREMMFF1

Display

DISPLAY		Display option	ISREMF1
PQUERY		Obtain panel information	ISREMF4
SETMSG		Set message display	ISREMF3
TBDISPL		Table display information	ISREMF2

Library Access

LMCLOSE		Close a data set	ISREMF1
LMERASE		Erase a data set or library	ISREMF2
LMFREE		Release a data set	ISREMF3
LMGET		Read a record	ISREMF4
LMINIT		Establish a data id	ISREMF5
LMMADD		Add a member	ISREMF6
LMMDEL		Delete a member	ISREMF7
LMMFIND		Find a member	ISREMF8
LMMLIST		Create a member list	ISREMF9
LMMREN		Rename a member	ISREMF1A
LMMREP		Replace a member	ISREMF1B
LMOPEN		Open a data set	ISREMF1C
LMPROM		Promote a member	ISREMF1D
LMPUT		Write a record	ISREMF1E
LMQUERY		Provide data set information	ISREMF1F
LMRENAME		Rename a library	ISREMF1G

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

Miscellaneous

BROWSE	MVS	Browse service (MVS)	ISREMF1
BROWSE	VM	Browse service (VM)	ISREMF2
BROWSE	VSE	Browse service (VSE)	ISREMF3
CONTROL		Control service	ISREMF4
EDIT	MVS	Edit service (MVS)	ISREMF5
EDIT	VM	Edit service (VM)	ISREMF6
EDIT	VSE	Edit service (VSE)	ISREMF7
GETMSG		Get message service	ISREMF8
LOG		Write message or log file	ISREMF9
SELECT		Select service	ISREMF10

MESSAGE FORMAT

MSG		Message member selection	ISREMF11
-----	--	--------------------------	----------

PANEL FORMATS AND STATEMENTS

Panel Formats

ENTRY		Data entry panel	ISREMF12
MULTIPLE		Multiple-column panel	ISREMF13
MULTIPLE	SELECT2	Double-column selection panel	ISREMF14
MULTIPLE	ENTRY2	Double-column entry panel	ISREMF15
SELECTION		Menu/selection panel	ISREMF16
TBDISPL		Table display table	ISREMF17
TUTORIAL		Help/tutorial panel	ISREMF18

Panel Statements

ASSIGN		Assignment statement	ISREMF19
ASSIGN	SIMPLE	Simple assignment statement	ISREMF20
ASSIGN	TRANS	Trans assignment statement	ISREMF21
ASSIGN	TRUNC	Trunc assignment statement	ISREMF22
ASSIGN	TRANSTRU	Nested translate truncate statement	ISREMF23
ATTR		Attribute section header	ISREMF24
ATTRIB		New attribute character definition	ISREMF25
ATTRIBA		New attrib char definition for area	ISREMF26
BODY		Body section header	ISREMF27
CONTROL		Control variables	ISREMF28
CONTROL	CURSOR	Control initial cursor placement	ISREMF29
CONTROL	HELP	Establish a tutorial panel	ISREMF30
CONTROL	MSG	Identify message to be displayed	ISREMF31
CONTROL	RESP	Indicate user response to panel	ISREMF32
CONTROL	TRAIL	Contain remainder from TRUNC function	ISREMF33
IF		IF statement	ISREMF34
MODEL		Model section header	ISREMF35
REFRESH		Refetch variables prior to redisplay	ISREMF36
VERIFY		Verification statements	ISREMF37

Model Name	Qualifier	Description	Member Name
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 PASCAL PROGRAMS

PASCAL Definitions

WORKSTOR	PASCAL Definitions	ISREMQPD
----------	--------------------	----------

Graphics

GRERROR	Graphics error block service	ISREMQS1
GRINIT	Graphics initialization	ISREMQS2
GRTERM	Graphics termination service	ISREMQS3

Table Functions (General)

TBCLOSE	Table close	ISREMQG5
TBCREATE	Table create	ISREMQG1
TBEND	Table end	ISREMQG6
TBERASE	Table erase	ISREMQG7
TBOPEN	Table open	ISREMQG2
TBQUERY	Table query	ISREMQG3
TBSAVE	Table save	ISREMQG4
TBSTATS	Table statistics	ISREMQG8

Table Functions (Row)

TBADD	Table row add	ISREMQR1
TBBOTTOM	Table row pointer to bottom	ISREMQRA
TBDELETE	Table delete	ISREMQR2
TBEXIST	Table exist	ISREMBR6
TBGET	Table get	ISREMBR3
TBPUT	Table put	ISREMBR4
TBMOD	Table modify	ISREMBR5
TBSARG	Table search argument	ISREMBR7
TBSCAN	Table scan	ISREMQR8
TBSKIP	Table skip	ISREMQRB
TBSORT	Table sort	ISREMQRD

Model Name	Qualifier	Description	Member Name
TBTOP		Table top	ISREMQR9
TBVCLEAR		Table variable clear	ISREMQRC

Variables

VCOPY		Copy variable	ISREMQV5
VDEFINE		Variable define	ISREMQV3
VDELETE		Variable delete	ISREMQV4
VGET		Variable get	ISREMQV1
VPUT		Variable put	ISREMQV2
VREPLACE		Variable replace	ISREMQV6
VRESET		Variable reset	ISREMQV7

File Tailoring

FTCLOSE		End file tailoring	ISREMQF3
FTERASE		File tailor erase	ISREMQF4
FTINCL		File tailor include skeleton	ISREMQF2
FTOPEN		File tailor open	IREMMQF1

Display

DISPLAY		Display option	ISREMQD1
PQUERY		Obtain panel information	ISREMQD4
SETMSG		Set message display	ISREMQD3
TBDISPL		Table display information	ISREMQD2

Library Access

LMCLOSE		Close a data set	ISREML1
LIMERASE		Erase a data set or library	ISREML2
LMFREE		Release a data set	ISREML3
LMGET		Read a record	ISREML4
LMINIT		Establish a data id	ISREML5
LMMADD		Add a member	ISREML6
LMMDEL		Delete a member	ISREML7
LMMFIND		Find a member	ISREML8
LMMLIST		Create a member list	ISREML9
LMMREN		Rename a member	ISREMLA
LMMREP		Replace a member	ISREMLB
LMOPEN		Open a data set	ISREMLC
LMPROM		Promote a member	ISREMLD
LMPUT		Write a record	ISREMLE
LMQUERY		Provide data set information	ISREMLF
LMRENAME		Rename a library	ISREMLG

Model Name	Qualifier	Description	Member Name
------------	-----------	-------------	-------------

Miscellaneous

BROWSE	MVS	Browse service (MVS)	ISREMQB1
BROWSE	VM	Browse service (VM)	ISREMQB2
BROWSE	VSE	Browse service (VSE)	ISREMQB3
CONTROL		Control service	ISREMQM2
EDIT	MVS	Edit service (MVS)	ISREMQE1
EDIT	VM	Edit service (VM)	ISREMQE2
EDIT	VSE	Edit service (VSE)	ISREMQE3
GETMSG		Get message service	ISREMQM6
LOG		Write message or log file	ISREMQM5
SELECT		Select service	ISREMQM1

ISPF SERVICES IN PL/I PROGRAMS

Graphics

GRERROR		Graphics error block service	ISREMP51
GRINIT		Graphics initialization	ISREMP52
GRTERM		Graphics termination service	ISREMP53

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
TBSTATS		Table statistics	ISREMPG8

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

Model Name	Qualifier	Description	Member Name
TBSKIP		Table skip	ISREMPRB
TBSORT		Table sort	ISREMPRD
TBTOP		Table top	ISREMPR9
TBVCLEAR		Table variable clear	ISREMPRC

Display

DISPLAY		Display option	ISREMPD1
PQUERY		Obtain panel information	ISREMPD4
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

Library Access

LMCLOSE		Close a data set	ISREMP11
LMERASE		Erase a data set or library	ISREMP12
LMFREE		Release a data set	ISREMP13
LMGET		Read a record	ISREMP14
LMINIT		Establish a data id	ISREMP15
LMADD		Add a member	ISREMP16
LMMDEL		Delete a member	ISREMP17
LMMFIND		Find a member	ISREMP18
LMMLIST		Create a member list	ISREMP19
LMREN		Rename a member	ISREMP1A
LMMREP		Replace a member	ISREMP1B
LMOPEN		Open a data set	ISREMP1C
LMPROM		Promote a member	ISREMP1D
LMPUT		Write a record	ISREMP1E
LMQUERY		Provide data set information	ISREMP1F
LMRENAME		Rename a library	ISREMP1G

Miscellaneous

BROWSE	MVS	Browse service (MVS)	ISREMPB1
BROWSE	VM	Browse service (VM)	ISREMPB2
BROWSE	VSE	Browse service (VSE)	ISREMPB3
CONTROL		Control service	ISREMPM2
EDIT	MVS	Edit service (MVS)	ISREMP1
EDIT	VM	Edit service (VM)	ISREMP2
EDIT	VSE	Edit service (VSE)	ISREMP3

Model Name	Qualifier	Description	Member Name
GETMSG		Get message service	ISREMPM6
LOG		Write message or log file	ISREMPM5
SELECT		SELECT service	ISREMPM1

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

IBM Corporation
International Business Machines Corporation
1972

IBM Corporation
International Business Machines Corporation
1972

APPENDIX G. LIST OF ABBREVIATIONS

The following list includes the command names and keywords that can be abbreviated, followed by the allowable abbreviation(s). It is recommended that, in order to improve readability, abbreviations not be used in edit macros.

AFTER	AFT			
BEFORE	BEF			
BOUNDS	BOUND	BNDS	BND	
BROWSE	BRO			
CANCEL	CAN			
CHANGE	CHA	CHG	C	
CHARS	CHAR			
COMMAND	CMD	COM		
COBOL	COB			
COLUMNS	COLS	COL		
CREATE	CRE			
CURSOR	CUR	CSR		
DATA	D			
DEFINE	DEF			
DELETE	DEL			
DISPLAY	DIS	DISP	DISPL	
ERR	ERROR			
EXCLUDED	EXCLUDE	EXC	EX	X
FIND	F			
HALF	H			
LABEL	LABELS	LAB		
LEVEL	LEV			
LOCATE	LOC	L		
MAX	M			
MODEL	MOD			
NOCOBOL	NOCOB			
NOCOLS	NOCOL			
NONOTES	NONOTE	NONOT		
NONULLS	NONULL	NONUL		
NONUM	NONUMBR	NONUMB	NONUMBER	
NOPROCESS	NOPROC			
NOTABS	NOTAB			
NOTES	NOTE			
NULLS	NULL	NUL		
NUMBER	NUMB	NUM		
NX	NONX			
PAGE	P			
PROGRAM	PGM			
PREFIX	PRE			
PROFILE	PROF	PRO	PR	
RECOVERY	RECOVER	RECOVRY	RECVRY	RECVR
RECOVERY	RECOV	REC		
RENUM	REN			

REPLACE	REPL	REP	
RESET	RES		
SPECIAL	SPE		
STANDARD	STD		
SUBMIT	SUB		
SUFFIX	SUF		
TABS	TAB		
UNNUMBER	UNNUMB	UNNUM	UNN
VERSION	VERS	VER	
VERTICAL	VERT		

INDEX

Special Characters

- < (shift data left) line command
edit 159
- ((shift columns left) line command
edit 159
-) (shift columns right) line command
edit 159
- > (shift data right) line command
edit 159

A

- A (after) line command 134
- accessing 3850 virtual volumes 32
- add row entry panel
dialog test 285
- add row tables option
dialog test 283
- adding dialog variables
dialog test 272
- adding lines 129
- adding models 174
- aliases for compatibility
previous releases 348
- allocating a new data set (data set
utility) 187
- ampersand (&) prefix
used to redisplay commands (edit) 66
- APL keyboard
internal character translation 331
- application command table 216
- assembler/compiler processing 234
- ATTENTION (PA1) key 17, 257
- AUTOLIST command 89
- autolist mode 89
- automatic source listing
edit 89
- AUTONUM command
edit 86
- autonum mode
edit 86
- AUTOSAVE command

- edit 87
- autosave mode
edit 87

B

- B (before) line command 134
- background
See batch
- batch (option 5) 249
- batch output listing
allocation 37
browsing 37
printing 37
- bottom
scrolling to 19
- boundaries, edit
controlling 96
DBCS data 124
- boundary definition line
edit 98, 158
- BOUNDS line command
edit 157
- BOUNDS primary command (edit) 96
- breakpoint primary option menu
dialog test 308
- breakpoint qualification panel
dialog test 303
- breakpoints entry panel
dialog test 302
- breakpoints option
dialog test 301
- browse (option 1) 53
BROWSE command 63
column identification line 58
COLUMNS command 58
command summary 335
commands
syntax 343
data set characteristics 53
DISPLAY command 61
FIND command 59
find/change strings 347
functions 57
HEX command 62

- label 57
- LOCATE command 57
- mixed data 54
- program function keys 18
- RESET command 58
- scale line 58
- setting hex mode 62
- browse data set
 - data set list utility (option 3.4) 204
- browse log option
 - dialog test 291
- browsing a member (library utility) 183
- built-in command
 - executing 104
- built-in labels 80
- BUILTIN command 104



- C (copy) line command
 - used with CREATE or REPLACE 121
- C (copy) line command 135
- CANCEL command
 - dialog test 264
 - edit 127
- cancelling
 - a macro definition 103
- CAPS command
 - DBCS data 90
 - edit 89
- caps mode
 - edit 89
- catalog data set
 - data set list utility (option 3.4) 204
- cataloging a data set (data set utility) 189
- chaining commands
 - See stacking commands
- CHANGE command
 - edit 106
- CHANGE PF key function
 - See RCHANGE command
- changing
 - indentations
 - edit 159
- changing data
 - edit 106
- changing dialog variables
 - dialog test 272

- changing models 174
- changing PF keys
 - KEYS command 16
- changing program function keys 45
- character strings (edit FIND or CHANGE) 114
- character translation
 - edit 89
- CLIST
 - attention exits 17, 257
 - ISPF services in 349
- closed table status display panel
 - dialog test 288
- COBOL
 - programs
 - ISPF services in 351
 - restrictions
 - member expansion 231
 - member parts list 231
- column identification line
 - browse 58
 - edit 152
- column limitation (edit FIND, CHANGE, and EXCLUDE) 116
- column shifting
 - DBCS data 163
- COLUMNS
 - line command
 - edit 152
 - primary command
 - browse 58
 - shifting 159
- command (option 6) 255
 - restrictions 256
- command and long message areas
 - user specification of location of 49
- command table
 - application 216
- command table utility (option 3.9) 216
 - line commands 217
 - primary commands 218
- command/option area 6
- commands
 - browse (option 1)
 - summary 335
 - syntax 343
 - command table utility (option 3.9)
 - line 217
 - primary 218
 - description summary 333
 - dialog test line 263, 265
 - summary 342
 - dialog test primary 263, 264
 - summary 341

- edit (option 2) line
 - summary 339
 - syntax 347
- edit (option 2) primary
 - summary 336
 - syntax 344
- entering 13
- general
 - description 14
 - PF keys assigned 14
 - summary 333
- member list
 - summary 335
 - syntax 343
- scrolling 19
- stacking 13, 42
- syntax
 - summary 343
- commands for compatibility
 - previous releases 348
- compatibility with prior releases
 - commands and aliases 348
- compiling a program
 - batch 249
 - foreground 229
- compress data set
 - data set list utility (option 3.4) 204
- compressing a data set (library utility) 183
- concatenating libraries 25
- controlling nulls
 - edit 90
- controlling the edit boundaries 96
- convert menus/messages utility (option 3.10) 219
- COPY command (edit) 117
- CREATE command
 - edit 120
- creating a new member 66
- creating or updating statistics (reset statistics utility) 206
- CURSOR command 15

D

- D (delete) line command
 - dialog test 265
 - edit 131
- data
 - shifting 159

- data set characteristics
 - browse 53
 - edit 65
- data set information
 - displaying (library utility) 184
- data set list
 - LONG format 203
 - QUICK format 203
 - SHORT format 203
 - specifying DSNNAME LEVEL 202
 - specifying list format 202
- data set list listing
 - format 324
- data set list utility (option 3.4) 201
 - commands
 - LOCATE 204
 - SORT 204
 - display data set list 201
 - print data set list 201
- data set password specification 32
- data set security 32
- data set utility (option 3.2) 186
- data shifting
 - DBCS data 161
- DBCS data
 - CAPS command 90
 - column shifting 163
 - data shifting 161
 - dialog text 275
 - display boundary 65
 - FIND command 60
 - FIND command (edit) 107
 - hardware tabs 92
 - RFIND 60
 - RFIND command (edit) 107
 - TF (text flow) line command 144
 - TS (text split) line command 140
- DEFINE command 103
- defining a name 103
- defining boundary columns
 - edit 157
- defining masks
 - edit 153
- defining program function keys 45
 - KEYS command 45
 - specifying line commands 46
 - specifying primary commands 46
- DELETE command (edit) 122
- delete data set
 - data set list utility (option 3.4) 204
- delete row tables option
 - dialog test 280
- delete table row confirmation panel

- dialog test 280
- deleting a member (library utility) 183
- deleting held job output (outlist utility) 214
- deleting lines 122, 131
- deleting models 174
- deleting statistics (reset statistics utility) 207
- delimiter
 - command stacking 13, 42
- DFDS 212
- dialog development 2
 - models 168, 349
- dialog test 1
- dialog test (option 7) 259
 - breakpoints option 301
 - browse log option 291
 - environment 260
 - functions option 266
 - invoke service option 294
 - line commands 263, 265
 - summary 342
 - panels option 269
 - primary commands 263, 264
 - summary 341
 - primary option menu 260
 - severe error handling 262
 - tables option 276
 - add row 283
 - delete row 280
 - display row 278
 - display status 287
 - display structure 285
 - modify row 281
 - traces option 296
 - variable usage 261
 - variables option 272
- dialog variables
 - manipulating 273
- display boundary
 - DBCS data 65
- DISPLAY command
 - browse 61
- display data set information
 - data set list utility (option 3.4) 204
- display data set list
 - data set list utility (option 3.4) 201
- display member list
 - data set list utility (option 3.4) 204
- display panel entry panel
 - dialog test 269

- display row tables option
 - dialog test 278
- display status tables option
 - dialog test 287
- display structure tables option
 - dialog test 285
- display table row entry panel
 - dialog test 279
- displaying a member list (library utility) 183
- displaying a message 270
- displaying a panel 269
- displaying data in hexadecimal format
 - browse 62
- displaying data set information (data set utility) 189, 190
- displaying data set information (library utility) 184
- displaying dialog variables
 - dialog test 272
- displaying held job output (outlist utility) 214
- displaying model notes 98
- displaying the panel identifier
 - PANELID command 16
- displaying VTOC entries (list VTOC utility) 211
- dit option 2/LC command 148
- dit option 2/UC command 146
- documentation preparation 1
- DOWN command 14
- DSNAME LEVEL
 - specifying 202
- DSPRINT TSO command processor 210
- duplicating lines 132

E

- edit (option 2) 65
 - < (shift data left) line command 159
 - ((shift columns left) line command 159
 -) (shift columns right) line command 159
 - > (shift data right) line command 159
 - A (after) line command 134
 - AUTOLIST command 89
 - autolist mode 89
 - AUTONUM command 86
 - autonum mode 86

AUTOSAVE command 87
 autosave mode 87
 B (before) line command 134
 boundary definition line 98, 158
 BOUNDS command 96
 BOUNDS line command 157
 BUILTIN 104
 C (copy) line command 135
 CANCEL command 127
 CAPS command 89
 caps mode 89
 CHANGE command 106
 changing indentations 159
 column identification line 152
 COLUMNS line command 152
 COPY command 117
 CREATE command 120
 D (delete) line command 131
 data set characteristics 65
 DEFINE command 103
 defining boundary columns 157
 defining masks 153
 DELETE command 122
 disabling library management 67
 EDIT command 125
 EXCLUDE command 106
 excluding lines 164
 F (first) line command 166
 FIND command 106
 find/change strings 347
 functions 72
 HEX command 95
 hex mode 95
 hexadecimal display 95
 I (insert) line command 129
 identifying columns 152
 IMACRO command 101
 L (last) line command 166
 LEVEL command 87
 line command summary 339
 line commands 73, 127
 syntax 347
 LOCATE command 104
 locating lines 104
 M (move) line command 134
 macros 77
 MASK line command 153
 merging data 117
 mixed data 69
 MODEL command 171
 models 168
 MOVE command 117
 NOTE command 98
 NULLS command 90
 nulls mode 90
 NUMBER command 84
 number mode 84
 O (overlay) line command 150
 overlaying lines 150
 PACK command 88
 preparing text 136
 primary command summary 336
 primary commands 73
 syntax 344
 PROFILE command 93
 profile display 93
 program function keys 18
 R (repeat) line command 132
 RECOVERY command 100
 recovery control 100
 recovery mode 100
 recovery panel 100
 RENUM command 85
 REPLACE command 120
 RESET command 93, 99
 RMACRO command 102
 S (show) line command 166
 SAVE command 126
 scale line 152
 segmenting data 117
 sequence number display 83
 sequence number format 83
 sequence number generation and
 control 82
 SORT command 123
 STATS command 86
 stats mode 86
 SUBMIT command 125
 tab definition line 156
 tabs control 91, 155
 tabs definition 91
 TABS line command 155
 tabs mode 92
 TABS primary command 92
 TE (text entry) line command 137
 termination 81
 TF (text flow) line command 143
 TS (text split) line command 138
 UNNUM command 85
 VERSION command 87
 X (exclude) line command 164
 edit data set
 data set list utility (option
 3.4) 204
 edit profile
 display and control 93
 specifying 69
 edit recovery panel 100

- edit recovery table 100
- editing a new member 66
- encountering a breakpoint
 - dialog test 308
- END command 14, 23
 - dialog test 264
- entering commands 13
- entry panel 6
- example of limiting operands (edit FIND, CHANGE, and EXCLUDE) 117
- EXCLUDE command
 - edit 106
- excluded line limitation (edit FIND and CHANGE) 116
- excluding data
 - edit 106
- excluding lines
 - edit 164
- EXECs
 - ISPF services in 353
- executing a built-in command 104
- expansion trigger 229
- extended return
 - See jump function

F

- F (first) line command
 - edit 166
- file tailoring control statements
 - models 361
- FIND command
 - browse 59
 - edit 106
- FIND PF key function
 - See RFIND command
- find/change strings
 - browse (option 1) 347
 - edit (option 2) 347
 - specification 111
- finding character strings
 - browse 59
- finding data
 - edit 106
- foreground (option 4) 229
 - processing sequence 233
- foreground output listing
 - allocation 37

- browsing 37
- printing 37
- format definition utility (Option 3.11) 222
- format of panels 5
- formatted data
 - format definition 222
 - MASK command 154
 - shifting data 160
 - TE (text entry) line command 137
- formatting a document
 - foreground 229
- FORTRAN
 - programs
 - ISPF services in 354
- function trace entry panel
 - dialog test 297
- functions option
 - dialog test 266

G

- general commands
 - summary 333
- generating JCL for printing or punching (hardcopy utility) 210
- getting message explanation 22

H

- hardcopy utility (option 3.6) 209
- hardware tabs
 - DBCS data 92
- hardware tabs (edit) 91
- HELP command 14, 22
 - tutorial (option T) 313
- help information 22
- HEX command
 - browse 62
 - edit 95
- hex mode
 - browse 62
 - edit 95
- hexadecimal display
 - edit 95

I

I (insert) line command
 dialog test 265
 edit 129
identifying columns
 browse 58
 edit 152
IMACRO command 101
indentations
 changing
 edit 159
index listing
 format 323
initial macro
 specifying 101
inserting lines 129
invocation of PDF 10
invoke dialog service entry panel
 dialog test 294
invoke function entry panel
 dialog test 267
invoke service option
 dialog test 294
invoking
 command procedure 266
 panel 266
 program 266
ISPCMDS 216
ISPF library 25
ISPF services
 in CLIST 349
 in COBOL programs 351
 in EXECs 353
 in FORTRAN programs 354
 in PASCAL programs 357
 in PL/I programs 359

J

JCL for printing or punching
 generating (hardcopy utility) 210
job statement specification 33
job statements
 in user profile 34
jump function 16, 23

K

KEYS command 16

L

L (last) line command
 edit 166
label
 browse 57
labels
 built-in 80
 edit 77
language processors 1
LC command 148
 example 148
LEFT command 15
LEVEL command
 edit 87
library access services 25
 models 349
library and data set name
 specification 31
library concatenation 25
library hierarchy 25
library management
 disabling
 edit 67
library statistics 27
 reset statistics utility 28
library utility (option 3.1) 182
line commands
 command table utility (option
 3.9) 217
 edit 127
 rules for entering 128
line numbers
 edit 82
line range
 edit 77
lines
 excluded 106, 164
 locating 105
 resetting 99
link editing a program
 batch 249
 foreground 229
linkage editor processing 237
list and log files

- format 323
- list file
 - description 36
- list file defaults
 - specifying
 - using parms option 39
- list VTOC utility (option 3.7) 211
- listing job names (outlist utility) 214
- listing the parts of a program
 - batch 249
 - foreground 229
- LOCATE command
 - data set list utility (option 3.4) 204
 - dialog test 264
 - edit 104
 - member list 35
 - under browse 57
- locating lines
 - browse 57
 - edit 104
- locating members 35
- locating models 180
- location of command and long message areas
 - user specification of 49
- log and list files 36
 - allocation 36
 - printing 37
 - setting the default allocation 44
 - specifying defaults 43
- log file
 - description 36
- log listing
 - format 324
- logical screens
 - switching between 21
- logical tabs (edit) 91
- LONG format
 - data set list 203
- long message and command areas
 - user specification of location of 49
- long message area 6
- lower case (LC) command
 - example 148
- lowercase (LC) command 148

M

- M (move) line command 134
 - used with CREATE or REPLACE 121
- macro name
 - defining 103
- macros
 - edit 77
- manipulating dialog variables 273
- MASK line command
 - edit 153
- member expansion 229
 - COBOL restrictions 231
 - restrictions 230
 - trigger statement errors 232
- member list 6, 34
 - browsing a member 36
 - commands 34
 - summary 335
 - syntax 343
 - deleting a member 36
 - LOCATE command 35
 - multiple-selection 36
 - printing a member 36
 - renaming a member 36
 - scrolling 35
 - SELECT command 35
 - selecting a member 36
 - selection codes 35, 36
 - single-selection 35
- member parts list
 - batch 252
 - COBOL restrictions 231
 - foreground 245
 - restrictions 230
 - trigger statement errors 232
- menu
 - See selection panel
- merging data (edit) 117
- message
 - long 6
 - short 5
- message display panel
 - dialog test 270
- message explanation 22
- message format models 356
- messages
 - displaying 270
- mixed data
 - browse 54
 - CHANGE strings 108
 - edit 69

- FIND command 60
- model class 169
- MODEL command
 - edit 171
- model name 169
- model name qualifier 169
- model notes
 - displaying 98
- model selection panel 173
- models 1, 2, 168, 349
 - adding 174
 - changing 174
 - class 169
 - deleting 174
 - dialog development 349
 - file tailoring control
 - statements 361
 - hierarchy 169
 - locating 180
 - logical name 169
 - message format 356
 - name 169
 - name qualifier 169
 - panel format and statement 356
 - types 169
 - using 172
- modification level 87, 207
- modify row tables option
 - dialog test 281
- modify table row entry panel
 - dialog test 282
- MOVE command (edit) 117
- move/copy utility (option 3.3) 192

N

- nondisplayable characters
 - edit 65
- nondisplayable data
 - dialog test 275, 282, 285, 293
- NOTE command, edit 98
- NULLS command
 - edit 90
- nulls mode
 - edit 90
- NUMBER command
 - edit 84
- number mode
 - edit 84

O

- O (overlay) line command
 - edit 150
- online tutorial 1
- open table status display panel
 - dialog test 289
- option X 23
- outlist utility (option 3.8) 213
- overlying lines
 - edit 150
- overtyping of parameters 20

P

- PACK command 88
- pack mode 88
- packing data
 - move/copy utility (option 3.3) 194
- pad character 41
- panel 5
 - add row entry
 - dialog test 285
 - breakpoint primary option menu
 - dialog test 308
 - breakpoint qualification
 - dialog test 303
 - breakpoints entry
 - dialog test 302
 - closed table status display
 - dialog test 288
 - command/option area 6
 - delete table row confirmation
 - dialog test 280
 - dialog test primary option menu 260
 - display panel entry
 - dialog test 269
 - display table row entry
 - dialog test 279
 - edit recovery 100
 - format 5
 - 3278 Model 5 5
 - format and statement models 356
 - function trace entry
 - dialog test 297
 - input field padding 41
 - invoke dialog service entry
 - dialog test 294
 - invoke function entry

- dialog test 267
- invoking
 - dialog test 266
- long message area 6
- message display
 - dialog test 270
- model selection 173
- modify table row entry
 - dialog test 282
- open table status display
 - dialog test 289
- parameter selection 39
- primary option menu 10
- scroll area 6
- short message area 5
- special display
 - dialog test 295
- table structure display
 - dialog test 286
- tables selection
 - dialog test 276
- terminal characteristics 40
- termination 23
- title area 5
- traces selection
 - dialog test 296
- TSO command entry 255
- types 6
- variable trace entry
 - dialog test 299
- variables entry
 - dialog test 272
- PANELID command 16
- panels
 - displaying 269
- panels option
 - dialog test 269
- parameter selection panel 39
- parameter specification 31
- parms option
 - used for specifying ISPF parameter options 39
- Parms option (option 0) 39
- PASCAL
 - programs
 - ISPF services in 357
- password protection 32
- PA1 key
 - See ATTENTION (PA1) key
- PA2 key
 - See RESHOW (PA2) key
- PDF 1
- PDF primary options 11
- PF keys
 - See also program function keys specifying
 - using parms option 39
 - picture string (edit FIND or CHANGE) 114
 - PL/I
 - programs
 - ISPF services in 359
 - prefixed commands
 - edit (option 2) 66
 - preparing text
 - edit 136
 - primary commands
 - command table utility (option 3.9) 218
 - primary option menu 10
 - PRINT command 15
 - print data set list
 - data set list utility (option 3.4) 201
 - print index listing
 - data set list utility (option 3.4) 204
 - print list
 - data set list utility (option 3.4) 204
 - PRINT-HI command 15
 - printer control characters
 - edit 65
 - printing a logical screen image 15, 334
 - printing a member (library utility) 183
 - printing a screen image 15
 - printing an entire data set (library utility) 183
 - printing an index listing (library utility) 183
 - printing held job output
 - outlist utility 214
 - printing VTOC entries (list VTOC utility) 211
 - PRINTL command 15, 334
 - PRINTLHI command 15, 334
 - PROFILE command
 - edit 93
 - profile display
 - edit 93
 - program access keys 17
 - ATTENTION (PA1) key 17
 - RESHOW (PA2) key 18
 - program function keys 13, 18
 - changing 45
 - defining 45
 - specifying 41, 45

Q

QUAL command
 dialog test 264
qualifying a breakpoint
 dialog test 303
QUICK format
 data set list 203

R

R (repeat) line command
 dialog test 265
 edit 132
range 79
 edit 77
range of lines limitation (edit FIND,
 CHANGE, and EXCLUDE) 117
RCHANGE command 14
RCHANGE PF key
 edit 109
RECOVERY command
 edit 100
recovery control
 edit 100
recovery macro
 specifying 102
recovery mode
 edit 100
redisplaying edit commands 66
removing lines 122
rename data set
 data set list utility (option
 3.4) 204
renaming a data set (data set
 utility) 188
renaming a member (library utility) 183
RENUM command
 edit 85
repeating lines 132
REPLACE command
 edit 120
requeuing held job output (outlist
 utility) 214
RESET command
 browse 58
 edit 93, 99
reset statistics utility (option
 3.5) 206

RESHOW (PA2) key 18
restrictions
 member expansion 230
 member parts list 230
RESUME command
 dialog test 264
RETURN command 14, 23
RFIND
 DBCS data 60
RFIND command 14
 with FIND command (browse) 60
RFIND PF key
 edit 109
RIGHT command 15

S

S (show) line command
 edit 166
SAVE command
 edit 126
saving data
 edit 126
scale line
 See column identification line
scroll area 6
scrollable data display 6
scrolling 1, 19
 commands 19
 DOWN command 14
 during browse 55
 LEFT command 15
 member list 35
 RIGHT command 15
 scroll amount 19
 to bottom 19
 to top 19
 UP command 14
 use of cursor 19
scrolling members 35
security
 data set 32
segmenting data (edit) 117
SELECT command
 member list 35
selection codes
 member list 35
selection panel 6
separator
 command stacking 13, 42
sequence number

- display 83
- format 83
- generation and control 82
- Session Manager 229, 257
- setting the edit boundaries 96
- shifting data 159
 - formatted data 160
- SHORT format
 - data set list 203
- short message area 5
- software tabs (edit) 91
- SORT command
 - data set list utility (option 3.4) 204
 - DBCS data 124
 - edit 123
- sorting data
 - edit 123
- special display panel
 - dialog test 295
- specifying
 - data set passwords 32
 - job statement information 33
 - library and data set names 31
 - log and list data set disposition at termination 23
 - pad character 41
 - parameters 31
 - PF keys 41
 - program function key definitions 45
 - screen format
 - 3278 Model 5 41
 - 3290 42
 - stacking delimiter 42
 - terminal characteristics 40
 - terminal type 40
- specifying a recovery macro (edit) 102
- specifying an initial macro (edit) 101
- specifying DSNNAME LEVEL
 - data set list utility (option 3.4) 202
- specifying ISPF parameter options
 - using parms option 39
- specifying list format
 - data set list utility (option 3.4) 202
- specifying log and list file
 - defaults 43
 - using parms option 39
- specifying screen display
 - characteristics (option 0.4) 49
- specifying terminal characteristics
 - using parms option 39
- SPLIT command 14
- split screen 1, 21
 - entering 21
 - for 3290 21
 - terminating 21
 - using the SPLIT command 21
 - using the SPLIT PF key 21
 - using the SWAP command 21
- stacking commands 13, 42
- statistics generation
 - edit 86
- STATS command
 - edit 86
- stats mode
 - edit 86
- structured programming 2
- SUBMIT command (edit) 125
- submitting to a job stream (edit) 125
- summary of command syntax 343
- summary of commands 333
- SWAP command 14
- switching between logical screens 21



- tab definition line
 - edit 156
- table structure display panel
 - dialog test 286
- tables option
 - dialog test 276
- tables selection panel
 - dialog test 276
- tabs control
 - edit 91, 155
- tabs definition
 - edit 91
- TABS line command
 - edit 155
- tabs mode
 - edit 92
- TABS primary command
 - edit 92
- TE (text entry) line command
 - edit 137
 - formatted data 137
- terminal characteristics
 - specifying 40
 - using parms option 39
- terminal characteristics panel 40
- terminal keys 17
- terminal type 40

- termination
 - edit 81
 - options 23
 - END command 23
 - jump function 23
 - option X 23
 - RETURN command 23
 - panel 23
 - PDF 23
 - specifying log and list file defaults 43
- testing dialogs 259
- TEXT keyboard
 - internal character translation 331
- text preparation
 - edit 136
- text strings (edit FIND or CHANGE) 114
- TF (text flow) line command
 - DBCS data 144
 - edit 143
- title area of panel 5
- top
 - scrolling to 19
- trace output 291
 - function trace entries 292
 - trace header 291
 - variable trace entries 292
- traces option
 - dialog test 296
- traces selection panel
 - dialog test 296
- translate
 - lowercase 148
 - uppercase 146
- trigger statement errors
 - member expansion 232
 - member parts list 232
- TS (text split) line command
 - DBCS data 140
 - edit 138
- TSO command 16
- TSO command entry panel 255
- tutorial (option T) 313
 - HELP command 313
- types of panels 6

U

- UC command 146
 - example 147
- uncatalog data set

- data set list utility (option 3.4) 204
- uncataloging a data set (data set utility) 189
- UNNUM command
 - edit 85
- UP command 14
- upper case (UC) command
 - example 147
- uppercase (UC) command 146
- usage scenario 3, 317
- user profile 33
 - scroll amount 20
- user specification of the location of the command and long message areas 49
- utilities
 - command table utility (option 3.9) 216
- utilities (option 3) 1, 181

V

- variable trace entry panel
 - dialog test 299
- variables entry panel
 - dialog test 272
- variables option
 - dialog test 272
 - usage 274
- VERSION command
 - edit 87
- version number 87, 207
- VTOC
 - display information (option 3.4) 205
 - displaying entries (list VTOC utility) 211
 - print information (option 3.4) 205
 - printing entries (list VTOC utility) 211

X

- X (exclude) line command
 - edit 164
- X option
 - used with jump function 17

Numerics

3850 virtual volumes
accessing 32

Interactive System Productivity Facility/
Program Development Facility
Version 2 Release 2 for MVS
Program Reference

SC34-4024-1

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.)

Cut or Fold Along Line

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 T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF/PDF for MVS

Program Reference

Printed in U.S.A.

SC34-4024-1



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.)

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 T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF/PDF for MVS

Program Reference

Printed in U.S.A.

SC34-4024-1



