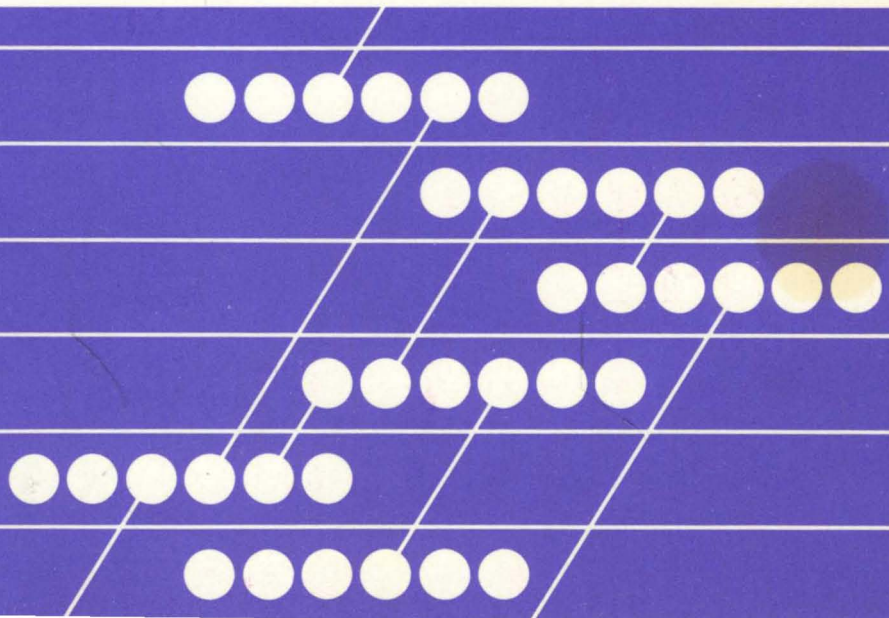


Interactive  
System Productivity Facility/  
Program Development Facility

Program Reference

MVS

**IBM**



---

Interactive  
System Productivity Facility/  
Program Development Facility

---

Program Reference

---

MVS

---

Publication Number  
SC34-2089-1

File Number  
S370/4300-39

---

Program Number  
5665-268

see SC34-2090  
for VM-version

**Second Edition (June, 1983)**

This major revision renders obsolete SC34-2089-0 and Technical Newsletter SN28-4602. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to the Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) Program Product, Program Number 5665-268, for use with Release 3.8 of OS/VS2 MVS, and to all subsequent releases until otherwise indicated by Technical Newsletters.

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

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 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, Publications, Dept. Z59, Bldg. 931, P.O. Box 390, Poughkeepsie, New York U.S.A. 12602. 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 -- use of the terminal, format of the displays, a description of the programming libraries, etc. Each succeeding chapter deals with the use of a particular PDF function, as selectable from the primary option menu. Finally, the appendixes contain a usage scenario, a description of listing formats, and command summaries.

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

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

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

ISPF for MVS Installation and Customization (SC34-2084) - Provides information needed to install ISPF and ISPF/PDF and to custom tailor these products for a particular installation.

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

- Uppercase commands and their operands should be entered as shown, but not necessarily in uppercase. Operands shown in lowercase are variable; you substitute your own value for them.
- Operands shown in brackets [ ] are optional, with a choice indicated by stacking the operands. You may choose one or none; the defaults are underscored.
- Operands shown without brackets are required. If several required operands are stacked, they are shown in braces { }; you must select one of the choices from the stack.
- Command name truncations are shown stacked under the full command name without braces.

# CONTENTS

|   |           |
|---|-----------|
| <b>Chapter 1. Introduction</b>                      | <b>1</b>  |
| Major Functions                                     | 1         |
| Structured Programming                              | 2         |
| Interactive Applications                            | 2         |
| Usage Scenario                                      | 3         |
| <b>Chapter 2. Interacting with PDF</b>              | <b>5</b>  |
| Panel Displays                                      | 5         |
| Format of Panels                                    | 5         |
| Types of Panels                                     | 6         |
| PDF Invocation                                      | 10        |
| Command Entry                                       | 13        |
| Jump Function                                       | 15        |
| Terminal Keys                                       | 16        |
| Program Access Keys                                 | 17        |
| Program Function Keys                               | 17        |
| Scrolling   | 18        |
| Overtyping of Scroll and Multiple Choice Parameters | 19        |
| Split Screen  | 20        |
| Help Information                                    | 21        |
| PDF Termination                                     | 22        |
| <b>Chapter 3. Libraries and Data Sets</b>           | <b>25</b> |
| ISPF Libraries                                      | 25        |
| Library Statistics                                  | 27        |
| Parameter Specification                             | 29        |
| Library and Data Set Names                          | 29        |
| Data Set Password                                   | 31        |
| Job Statement Information                           | 31        |
| User Profiles                                       | 32        |
| Member List Commands                                | 33        |
| Locating Members                                    | 33        |
| Single-Selection Member Lists                       | 33        |
| Multiple-Selection Member Lists                     | 34        |
| Log and List Files                                  | 35        |
| Foreground and Batch Output Listings                | 36        |
| Other Temporary Files                               | 36        |
| <b>Chapter 4. ISPF Parms (Option 0)</b>             | <b>37</b> |
| Specify Terminal Characteristics (Option 0.1)       | 37        |
| Specify Log and List Defaults (Option 0.2)          | 39        |
| Specify Program Function Keys (Option 0.3)          | 41        |
| <b>Chapter 5. Browse (Option 1)</b>                 | <b>45</b> |
| Locating Lines                                      | 47        |
| Identifying Columns                                 | 48        |
| Finding Character Strings                           | 49        |
| Translating Character Strings                       | 51        |

|  |           |
|--|-----------|
| Displaying Data in Hexadecimal Format . . . . .                  | 51        |
| <b>Chapter 6. Edit (Option 2)</b> . . . . .                      | <b>53</b> |
| Edit Selection . . . . .   | 54        |
| Summary of Edit Operations . . . . .                             | 57        |
| Command Functions . . . . .                                      | 57        |
| Edit Modes . . . . .   | 58        |
| Edit Profiles . . . . .  | 59        |
| Edit Termination . . . . .                                       | 61        |
| Primary Commands and Functions . . . . .                         | 61        |
| Controlling Your Edit Environment . . . . .                      | 62        |
| Sequence Number Generation and Control . . . . .                 | 62        |
| Sequence Number Format . . . . .                                 | 62        |
| Sequence Number Display . . . . .                                | 63        |
| Initialization of Number Mode . . . . .                          | 64        |
| NUMBER Command . . . . .   | 64        |
| RENUM Command . . . . .  | 65        |
| UNNUM Command . . . . .  | 65        |
| AUTONUM Command . . . . .  | 65        |
| Statistics Generation (STATS Command) . . . . .                  | 66        |
| Automatic Source Listing (AUTOLIST Command) . . . . .            | 66        |
| Character Translation (CAPS Command) . . . . .                   | 66        |
| Nulls Control (NULLS Command) . . . . .                          | 67        |
| Tabs Definition (TABS Primary Command) . . . . .                 | 68        |
| Profile Display and Control (PROFILE Command) . . . . .          | 70        |
| Hexadecimal Display (HEX Command) . . . . .                      | 71        |
| General Reset (RESET Command) . . . . .                          | 73        |
| Recovery Control (RECOVERY Command) . . . . .                    | 73        |
| Locating Lines (LOCATE Command) . . . . .                        | 74        |
| Finding and Changing Data (FIND and CHANGE Commands) . . . . .   | 75        |
| Command Summary . . . . .  | 75        |
| Detailed Command Description . . . . .                           | 78        |
| Starting Point, Direction, and Extent of Search . . . . .        | 80        |
| Conditions for Character String Match . . . . .                  | 80        |
| Use of Text Strings . . . . .                                    | 81        |
| Use of Picture Strings . . . . .                                 | 81        |
| Range and Column Limitations . . . . .                           | 82        |
| Merging and Segmenting Data . . . . .                            | 83        |
| Merging Data (COPY and MOVE Commands) . . . . .                  | 83        |
| Segmenting Data (CREATE and REPLACE Commands) . . . . .          | 85        |
| Submitting to a Job Stream (SUBMIT Command) . . . . .            | 87        |
| Saving and Cancelling . . . . .                                  | 88        |
| Saving Data (SAVE Command) . . . . .                             | 88        |
| Cancelling Changes (CANCEL Command) . . . . .                    | 89        |
| Line Commands and Functions . . . . .                            | 89        |
| Rules for Entering Line Commands . . . . .                       | 90        |
| Basic Line Commands . . . . .                                    | 91        |
| I (Insert) Command . . . . .                                     | 91        |
| D (Delete) Command . . . . .                                     | 92        |
| R (Repeat) Command . . . . .                                     | 93        |
| M (Move), C (Copy), A (After), and B (Before) Commands . . . . . | 95        |
| Preparing Text . . . . .   | 97        |
| TE (Text Entry) Command . . . . .                                | 98        |
| TS (Text Split) Command . . . . .                                | 99        |

|  |            |
|--|------------|
| TF (Text Flow) Command . . . . .                     | 102        |
| Overlaying Lines (O Command) . . . . .               | 104        |
| Identifying Columns (COLS Command) . . . . .         | 106        |
| Defining Masks (MASK Command) . . . . .              | 107        |
| Controlling Tabs (TABS Line Command) . . . . .       | 109        |
| Defining Boundary Columns (BOUNDS Command) . . . . . | 111        |
| Changing Indentations . . . . .                      | 113        |
| > and < (Data Shifting) Commands . . . . .           | 114        |
| ) and ( (Column Shifting) Commands . . . . .         | 116        |
| Excluding Lines (X, F, L, and S Commands) . . . . .  | 116        |
| Dialog Development Models . . . . .                  | 121        |
| MODEL Command . . . . .                              | 123        |
| Using Models . . . . .                               | 124        |
| Adding, Changing, and Deleting Models . . . . .      | 126        |
| Adding Models . . . . .                              | 127        |
| Finding Models . . . . .                             | 132        |
| Changing Models . . . . .                            | 132        |
| Deleting Models . . . . .                            | 132        |
| <b>Chapter 7. Utilities (Option 3) . . . . .</b>     | <b>133</b> |
| Library Utility (Option 3.1) . . . . .               | 134        |
| Data Set Utility (Option 3.2) . . . . .              | 137        |
| Move/Copy Utility (Option 3.3) . . . . .             | 141        |
| Catalog Management Utility (Option 3.4) . . . . .    | 147        |
| Reset Statistics Utility (Option 3.5) . . . . .      | 148        |
| Hardcopy Utility (Option 3.6) . . . . .              | 150        |
| List VTOC Utility (Option 3.7) . . . . .             | 152        |
| Outlist Utility (Option 3.8) . . . . .               | 154        |
| Command Table Utility (Option 3.9) . . . . .         | 157        |
| Convert Menus/Messages (Option 3.10) . . . . .       | 159        |
| <b>Chapter 8. Foreground (Option 4) . . . . .</b>    | <b>163</b> |
| Processing Sequence . . . . .                        | 164        |
| Assembler/Compiler Processing . . . . .              | 165        |
| Linkage Editor Processing . . . . .                  | 168        |
| SCRIPT/VS Processing . . . . .                       | 170        |
| Draft Document Options . . . . .                     | 171        |
| Final Document Options . . . . .                     | 173        |
| <b>Chapter 9. Batch (Option 5) . . . . .</b>         | <b>175</b> |
| <b>Chapter 10. Command (Option 6) . . . . .</b>      | <b>179</b> |
| <b>Chapter 11. Dialog Test (Option 7) . . . . .</b>  | <b>181</b> |
| Dialog Test Environment . . . . .                    | 182        |
| Dialog Test Variable Usage . . . . .                 | 183        |
| Dialog Test Severe Error Handling . . . . .          | 184        |
| Dialog Test Commands . . . . .                       | 185        |
| Primary Commands . . . . .                           | 186        |
| Line Commands . . . . .                              | 187        |
| Functions (Option 7.1) . . . . .                     | 187        |
| Panels (Option 7.2) . . . . .                        | 189        |
| Variables (Option 7.3) . . . . .                     | 191        |
| Manipulating Variables . . . . .                     | 193        |



|  |            |
|--|------------|
| Variables Usage Notes . . . . .                                      | 194        |
| Tables (Option 7.4) . . . . .  | 195        |
| Display Row (Tables Option 1) . . . . .                              | 197        |
| Delete Table Row (Tables Option 2) . . . . .                         | 199        |
| Modify Row (Tables Option 3) . . . . .                               | 200        |
| Add After Row (Tables Option 4) . . . . .                            | 202        |
| Display Structure (Tables Option 5) . . . . .                        | 204        |
| Display Status (Tables Option 6) . . . . .                           | 205        |
| Log (Option 7.5) . . . . .   | 208        |
| Trace Header . . . . .   | 209        |
| Function Trace Entries . . . . .                                     | 209        |
| Variable Trace Entries . . . . .                                     | 210        |
| Dialog Services (Option 7.6) . . . . .                               | 211        |
| Traces (Option 7.7) . . . . .  | 212        |
| Function Trace (Trace Option 1) . . . . .                            | 213        |
| Variable Trace (Trace Option 2) . . . . .                            | 215        |
| Breakpoints (Option 7.8) . . . . .                                   | 217        |
| Defining Breakpoints (Breakpoints Panel) . . . . .                   | 218        |
| Qualifying a Breakpoint (Qualification Panel) . . . . .              | 219        |
| Tutorial (Option 7.T) . . . . .                                      | 221        |
| Exit (Option 7.X) . . . . .  | 222        |
| Encountering a Breakpoint . . . . .                                  | 222        |
| <b>Chapter 12. Tutorial (Option T) . . . . .</b>                     | <b>225</b> |
| <b>Appendix A. Usage Scenario . . . . .</b>                          | <b>229</b> |
| <b>Appendix B. PDF Listing Formats . . . . .</b>                     | <b>235</b> |
| <b>Appendix C. Character Translations for APL and Text Keyboards</b> | <b>241</b> |
| <b>Appendix D. Summary of Commands . . . . .</b>                     | <b>243</b> |
| General Commands . . . . .   | 243        |
| Member List Commands . . . . .                                       | 244        |
| Browse Commands . . . . .  | 245        |
| Edit Primary Commands . . . . .                                      | 245        |
| Edit Line Commands . . . . .   | 247        |
| Dialog Test Primary Commands . . . . .                               | 250        |
| Dialog Test Line Commands . . . . .                                  | 250        |
| <b>Appendix E. Summary of Command Syntax . . . . .</b>               | <b>251</b> |
| Member List Commands . . . . .                                       | 251        |
| Browse Commands . . . . .  | 251        |
| Edit Primary Commands . . . . .                                      | 251        |
| Find/Change Strings (Browse and Edit) . . . . .                      | 253        |
| Edit Line Commands . . . . .   | 254        |
| <b>Appendix F. Dialog Development Model Listing . . . . .</b>        | <b>255</b> |
| <b>Index . . . . .</b>   | <b>263</b> |

## FIGURES

|     |   |     |
|-----|---|-----|
| 1.  | Utility Selection Panel . . . . .                         | 7   |
| 2.  | Browse - Entry Panel . . . . .                            | 8   |
| 3.  | Browse - Member List for a VSE Type SL Library . . . . .  | 9   |
| 4.  | Browse - Data Display . . . . .                           | 10  |
| 5.  | Primary Option Menu . . . . .                             | 11  |
| 6.  | Default Program Key Arrangement . . . . .                 | 18  |
| 7.  | Split Screen Example . . . . .                            | 21  |
| 8.  | Termination Panel . . . . .                               | 23  |
| 9.  | ISPF Library Organization . . . . .                       | 26  |
| 10. | Hierarchy of ISPF Libraries . . . . .                     | 27  |
| 11. | Member List Display . . . . .                             | 29  |
| 12. | Parameter Selection Panel . . . . .                       | 37  |
| 13. | Terminal Characteristics Panel . . . . .                  | 38  |
| 14. | Log and List Defaults Panel . . . . .                     | 40  |
| 15. | PF Key Definition Panel (12 PF Keys) . . . . .            | 42  |
| 16. | PF Key Definition Panels (24 PF Keys) . . . . .           | 43  |
| 17. | Browse - Entry Panel . . . . .                            | 45  |
| 18. | Browse - Data Display . . . . .                           | 46  |
| 19. | Browse - Column Identification Line . . . . .             | 49  |
| 20. | Browse - Hexadecimal Display, Vertical . . . . .          | 52  |
| 21. | Edit - Creating New Data . . . . .                        | 54  |
| 22. | Edit - Entry Panel . . . . .                              | 55  |
| 23. | Edit - Member List . . . . .                              | 56  |
| 24. | Edit - Data Display . . . . .                             | 57  |
| 25. | Edit - Caution Message . . . . .                          | 60  |
| 26. | Edit - Profile Display . . . . .                          | 71  |
| 27. | Edit - Hexadecimal Display, Data Representation . . . . . | 72  |
| 28. | Edit - Recovery Panel . . . . .                           | 74  |
| 29. | Edit - CHANGE ALL Command Example . . . . .               | 77  |
| 30. | Edit - COPY Command Example . . . . .                     | 84  |
| 31. | Edit - REPLACE Command Example . . . . .                  | 86  |
| 32. | Edit - Insert and Delete Lines Example . . . . .          | 92  |
| 33. | Edit - Repeat Line Example . . . . .                      | 94  |
| 34. | Edit - Copy Lines Example . . . . .                       | 96  |
| 35. | Edit - Text Split Example . . . . .                       | 99  |
| 36. | Edit - Move Sentence Example . . . . .                    | 101 |
| 37. | Edit - Text Restructure Example . . . . .                 | 103 |
| 38. | Edit - Overlay Line Example . . . . .                     | 105 |
| 39. | Edit - Column Identification Line . . . . .               | 107 |
| 40. | Edit - Define Mask Example . . . . .                      | 108 |
| 41. | Edit - Tab Definition Line . . . . .                      | 111 |
| 42. | Edit - Default Column Boundary Settings . . . . .         | 112 |
| 43. | Edit - Boundary Definition Line . . . . .                 | 113 |
| 44. | Edit - Left Shift Data Example . . . . .                  | 115 |
| 45. | Edit - Excluded Lines Example . . . . .                   | 118 |
| 46. | Edit - Show Lines Example . . . . .                       | 120 |
| 47. | Edit - Model Class Selection Panel . . . . .              | 123 |
| 48. | Edit - Model Selection Panel for Command Models . . . . . | 125 |

|      |  |     |
|------|--|-----|
| 49.  | Edit - Sample Dialog Development Model                       | 126 |
| 50.  | Edit - Sample Block Letter Model                             | 127 |
| 51.  | Edit - Panel Model Selection Panel                           | 128 |
| 52.  | Edit - Modified Visible Panel Model Selection Panel          | 129 |
| 53.  | Edit - Modified )PROC Section of Panel Model Selection Panel | 130 |
| 54.  | Edit - Block Letter Model Selection Panel Coding             | 131 |
| 55.  | Utility Selection Panel                                      | 133 |
| 56.  | Library Utility Panel  | 134 |
| 57.  | Library Utility - Print, Rename and Delete                   | 136 |
| 58.  | Data Set Utility Panel                                       | 137 |
| 59.  | Data Set Utility - Allocate New Data Set                     | 138 |
| 60.  | Data Set Utility - Rename Data Set                           | 140 |
| 61.  | Data Set Utility - Display Information                       | 141 |
| 62.  | Move/Copy Utility Panels                                     | 142 |
| 63.  | Move/Copy Utility - Copy Members from List                   | 144 |
| 64.  | Catalog Management Utility Panel                             | 148 |
| 65.  | Reset Statistics Utility Panel                               | 149 |
| 66.  | Hardcopy Utility Panel                                       | 151 |
| 67.  | List VTOC Utility Panel                                      | 153 |
| 68.  | Display VTOC Example   | 154 |
| 69.  | Outlist Utility Panel  | 155 |
| 70.  | Outlist Print Panel  | 156 |
| 71.  | Command Table Utility Panel                                  | 157 |
| 72.  | Command Table Editing Panel                                  | 158 |
| 73.  | "From" Panel for Converting Menu/Message Definitions         | 160 |
| 74.  | "To" Panel for Converting Menu/Message Definitions           | 161 |
| 75.  | Foreground Selection Panel                                   | 163 |
| 76.  | Foreground - Assembler Example                               | 164 |
| 77.  | Foreground Print Panel                                       | 166 |
| 78.  | Foreground - Linkage Editor Example                          | 169 |
| 79.  | SCRIPT/VS Utility - First Panel                              | 171 |
| 80.  | SCRIPT/VS Draft Document Options                             | 172 |
| 81.  | SCRIPT/VS Final Document Options                             | 173 |
| 82.  | Batch Selection Panel  | 175 |
| 83.  | Batch - PL/I Optimizing Compiler Example                     | 176 |
| 84.  | Dialog Test - Primary Option Menu                            | 182 |
| 85.  | Dialog Test - Invoke Function Entry Panel                    | 188 |
| 86.  | Dialog Test - Display Panel Entry Panel                      | 189 |
| 87.  | Dialog Test - Message Display Panel                          | 191 |
| 88.  | Dialog Test - Variables Entry Panel                          | 192 |
| 89.  | Dialog Test - Tables Selection Panel                         | 195 |
| 90.  | Dialog Test - Display Table Row Entry Panel                  | 197 |
| 91.  | Dialog Test - Delete Table Row Confirmation Panel            | 199 |
| 92.  | Dialog Test - Modify Table Row Entry Panel                   | 200 |
| 93.  | Dialog Test - Add After Row Entry Panel                      | 203 |
| 94.  | Dialog Test - Table Structure Display                        | 205 |
| 95.  | Dialog Test - Table Status Display for a Closed Table        | 206 |
| 96.  | Dialog Test - Table Status Display for an Open Table         | 208 |
| 97.  | Dialog Test - Invoke Dialog Service Entry Panel              | 211 |
| 98.  | Dialog Test - Special Display Panel                          | 212 |
| 99.  | Dialog Test - Traces Selection Panel                         | 213 |
| 100. | Dialog Test - Function Trace Entry Panel                     | 214 |
| 101. | Dialog Test - Variable Trace Entry Panel                     | 216 |
| 102. | Dialog Test - Breakpoints Entry Panel                        | 219 |

|      |  |     |
|------|--|-----|
| 103. | Dialog Test - Qualification Panel . . . . .                | 220 |
| 104. | Dialog Test - Breakpoint Primary Option Menu . . . . .     | 222 |
| 105. | Tutorial - First Two Pages . . . . .                       | 225 |
| 106. | Sample Source Listing . . . . .                            | 237 |
| 107. | Sample Index Listing - Source Library . . . . .            | 238 |
| 108. | Sample Index Listing - Load Library . . . . .              | 239 |
| 109. | Sample Log Listing . . . . .                               | 240 |
| 110. | Internal Character Representations for APL Keyboards . . . | 241 |
| 111. | Internal Character Representations for Text Keyboards . .  | 242 |



## CHAPTER 1. INTRODUCTION

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

### MAJOR FUNCTIONS

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

- **Multilevel programming library support** that makes it easier to maintain and track program segments at different versions or levels.
- **Full-screen, context editing** that allows multiple changes to information on a screen with one interaction with the host system. The most frequently used editing functions are invoked using simple, one-character commands.
- **Models** that assist in the development of dialog panels, messages, function routines, file skeletons, and tables.
- **Scrolling** of source data and listings in any direction. Also, location of data by character string or line number.
- **Split screen** capability that allows you to partition the display screen into two "logical" areas.
- **Utilities** to specify and maintain libraries and 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.

#### 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.

## 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 or copy members or data sets
  4 CATALOG  - Catalog management:
                Display or print catalog entries
                Initialize or delete user catalog alias
  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
```

Figure 1. Utility Selection Panel

---

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

ISPF LIBRARY:  
PROJECT ==> SPFDEMO  
LIBRARY ==> 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)

Figure 2. Browse - Entry Panel

---

---

```

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

```

Figure 3. Browse - Member List

---

---

```

BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- LINE 000000 COL 001 080
COMMAND ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**
COINS: 00010001
PROCEDURE OPTIONS (MAIN); 00020000
  DECLARE 00030000
    COUNT FIXED BINARY (31) AUTOMATIC INIT (1), 00040000
    HALVES FIXED BINARY (31), 00050000
    QUARTERS FIXED BINARY (31), 00060003
    DIMES FIXED BINARY (31), 00070000
    NICKELS FIXED BINARY (31), 00080004
    SYSPRINT FILE STREAM OUTPUT PRINT; 00090000
  DO HALVES = 100 TO 0 BY -50; 00100000
  DO QUARTERS = (100 - HALVES) TO 0 BY -25; 00110000
  DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10; 00120000
  NICKELS = 100 - HALVES - QUARTERS - DIMES; 00130000
  PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS); 00140000
  COUNT = COUNT + 1; 00150000
  END; 00160000
  END; 00170000
  END; 00180000
END COINS; 00190001
***** BOTTOM OF DATA *****-CAPS ON-**

```

Figure 4. Browse - Data Display

---

## PDF INVOCATION

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 ==> _
                                USERID  - FLAG3
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
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, or 7, 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 or copy data; display or print catalog listings or VTOCs; reset library statistics; initiate hardcopy output; examine held SYSOUT data; create or modify application command tables; or convert menus and messages to ISPF format from old SPF format.
- 4 FOREGROUND** To execute language processing programs in the foreground, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL/VS, linkage editor, 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.
- C CHANGES** To obtain online instruction about the changes between the System Productivity Facility program development facility and ISPF/Program Development Facility.
- T TUTORIAL** To obtain immediate online instruction in the use of PDF. The tutorial may be viewed sequentially from beginning to end, or randomly by selecting topics from its table of contents or index. The tutorial may also be entered from other PDF options using the HELP command.
- X EXIT** To exit from PDF using the 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 keying the command in the command field and pressing ENTER.  
(This includes the command field in browse, edit, member lists, and table displays, as well as the command/option field on a panel.)
2. By pressing a program function (PF) key.

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 ignored or will be flagged as an invalid command. Any stacked valid commands in the command field will be executed.

You can stack commands for execution by entering a special delimiter between the commands. The default delimiter is a semicolon (;). You can change the default using the ISPF parms option (option 0.1). In the following example, two edit CHANGE commands have been stacked:

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

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

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

plus the following new commands: TSO, KEYS, and PANELID.

**Note:** The command designations for the FIND/CHANGE "key" functions have been changed to RFIND (repeat find) and RCHANGE (repeat change), to avoid confusion with the FIND and CHANGE commands that are used in browse and edit.

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

- HELP (PF1/13)** Displays additional information about an error message or provides tutorial information about PDF commands and options.
- SPLIT (PF2/14)** Causes split screen mode to be entered, or changes the location of the split line.
- END (PF3/15)** Terminates the current operation and returns to the previous panel. If the primary option menu is displayed, this command terminates PDF.
- RETURN (PF4/16)** Causes an immediate return to the primary option menu or to the panel from which a HELP or KEYS command was entered, without displaying the intervening panels. Note that the dialog test primary option menu (displayed when option 7 is selected) is itself a primary option menu; when you are in option 7, the RETURN command stops at that menu. For other uses of the RETURN command, see the section entitled "Jump Function."
- RFIND (PF5/17)** Repeats the action of the previous FIND command or the FIND part of the most recent CHANGE command (applies to browse and edit only).
- RCHANGE (PF6/18)** Repeats the action of the previous CHANGE command (applies to edit only).
- UP (PF7/19)** Causes a scroll toward the top of the data.
- DOWN (PF8/20)** Causes a scroll toward the bottom of the data.
- SWAP (PF9/21)** Moves the cursor to wherever it was previously positioned on the other logical screen of a split screen pair.
- LEFT (PF10/22)** Causes a scroll toward the left margin of the data.
- RIGHT (PF11/23)** Causes a scroll toward the right margin of the data.
- CURSOR (PF12/24)** Moves the cursor to the first input field on line 2 (normally the option selection or command field). Entering this command again causes the cursor to be moved to the second input field on line 2, if any (normally the scroll amount field).

- PRINT** Causes a "snapshot" of the screen image to be recorded in the ISPF list file, for subsequent printing.
- PRINT-HI** Same as PRINT except that high-intensity characters on the screen are printed with overstrikes to simulate the dual-intensity display. Do not use this command if the list output is to be 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. 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 eight 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 "CUR". 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 displayable input field preceded by an arrow (==>), enter an equal sign (=) immediately followed by a primary option. Then press the ENTER key or the RETURN PF key.

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

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

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

Note the following points:

- In the previous SPF product, this facility was called "extended return" and required the use of the RETURN PF key. For compatibility, you may still use the RETURN PF key when you enter the jump function, but use of the ENTER key is recommended.
- You may enter "=X", which is equivalent to selecting the X (exit) option on the primary option menu. Except under option 7, "=X" immediately terminates:
  - ISPF/PDF if you are not in split screen mode
  - The logical screen if you are in split screen mode

Commands stacked after =X in split screen mode will be ignored.

- 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.
- 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, this key should not be used while you are in ISPF full screen mode. See the discussion below for exceptions.

**RESHOW (PA2)** Redisplays the contents of the screen. 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<br>ATTENTION | PA2<br>RESHOW | PF1 / 13<br>HELP   | PF2 / 14<br>SPLIT  | PF3 / 15<br>END     |
|                  |               | PF4 / 16<br>RETURN | PF5 / 17<br>RFIND  | PF6 / 18<br>RCHANGE |
|                  |               | PF7 / 19<br>UP     | PF8 / 20<br>DOWN   | PF9 / 21<br>SWAP    |
|                  |               | PF10 / 22<br>LEFT  | PF11 / 23<br>RIGHT | PF12 / 24<br>CURSOR |

Figure 6. Default Program Key Arrangement

## SCROLLING

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

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

- A number from 1 to 9999 - specifies the number of lines (up or down) or columns (left or right) to be scrolled.
- PAGE - specifies scrolling by one page.
- HALF - specifies scrolling by a half page.
- MAX - specifies scrolling to the top, bottom, left margin, or right margin, depending upon which scrolling command is used.

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

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

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

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

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

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

and press the ENTER key, or enter

```
COMMAND ==> 3
```

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

## OVERTYPING OF SCROLL AND MULTIPLE CHOICE PARAMETERS

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

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

The same rule applies to multiple choice parameters 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 partition the display screen into two logical screens. The top and bottom screens are treated as though they were displayed on independent terminals. Functions that can be performed simultaneously on two separate terminals can be done on the top and bottom screens. Functions that are in conflict on two separate terminals (editing the same 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 line for the split and then pressing the SPLIT PF key. Alternatively, you can type SPLIT in the command/option field and then move the cursor to the desired line before pressing the ENTER key. In either case, the line containing the cursor becomes the split line and is identified by a row of periods. All lines below the cursor are treated as the bottom screen. The first display to appear on the bottom screen is always the primary option menu. An example is shown in Figure 7.

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

In split screen mode, 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.

```

BROWSE - SPFDemo.MYLIB.PLI(COINS) - 01.04 ----- LINE 000000 COL 001 080
COMMAND ==>                               SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-***
COINS:                                     00010001
  PROCEDURE OPTIONS (MAIN);                00020000
    DECLARE                                00030000
      COUNT  FIXED BINARY (31) AUTOMATIC INIT (1), 00040000
    . . . . .
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> _

          USERID  - BECKT
0  ISPF PARS - 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
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

## 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. 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 ==> J  
SYSOUT class ==> A  
Local printer ID ==>

LIST OPTIONS FOR THIS SESSION  
-----

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

VALID PROCESS OPTIONS:

J - Submit job to print (and delete)      K - Keep data set (do not print)  
L - Route to local printer (and delete)    D - Delete data set (do not print)

INSTRUCTIONS:

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

JOB STATEMENT INFORMATION: (If option "J" selected)

==> //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.library-name.type'
```

where

- "project-name" is the common identifier for all libraries belonging to the same project.
- "library-name" identifies the particular set of libraries, such as MASTER, TEST, or 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 version control over the programming development process and to reduce contention in library usage. Figure 10 shows a sample 3-level hierarchy consisting of a set of master libraries, a set of test libraries, and three sets of development (private) libraries identified by user id.

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

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

| ASM         | COBOL       | OBJ              | LOAD           | SCRIPT        |
|-------------|-------------|------------------|----------------|---------------|
| MEMBER<br>A | MEMBER<br>A | MEMBER<br>D      | MEMBER<br>IJK  | MEMBER<br>D   |
| MEMBER<br>B | MEMBER<br>D |                  |                |               |
| MEMBER<br>C | MEMBER<br>E | MEMBER<br>F      | MEMBER<br>PQRS | MEMBER<br>Q   |
|             | MEMBER<br>G | •<br>•<br>•<br>• |                | MEMBER<br>XYZ |
| •<br>•<br>• | •<br>•      |                  | •<br>•<br>•    | •<br>•<br>•   |

Figure 9. ISPF Library Organization

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.

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).

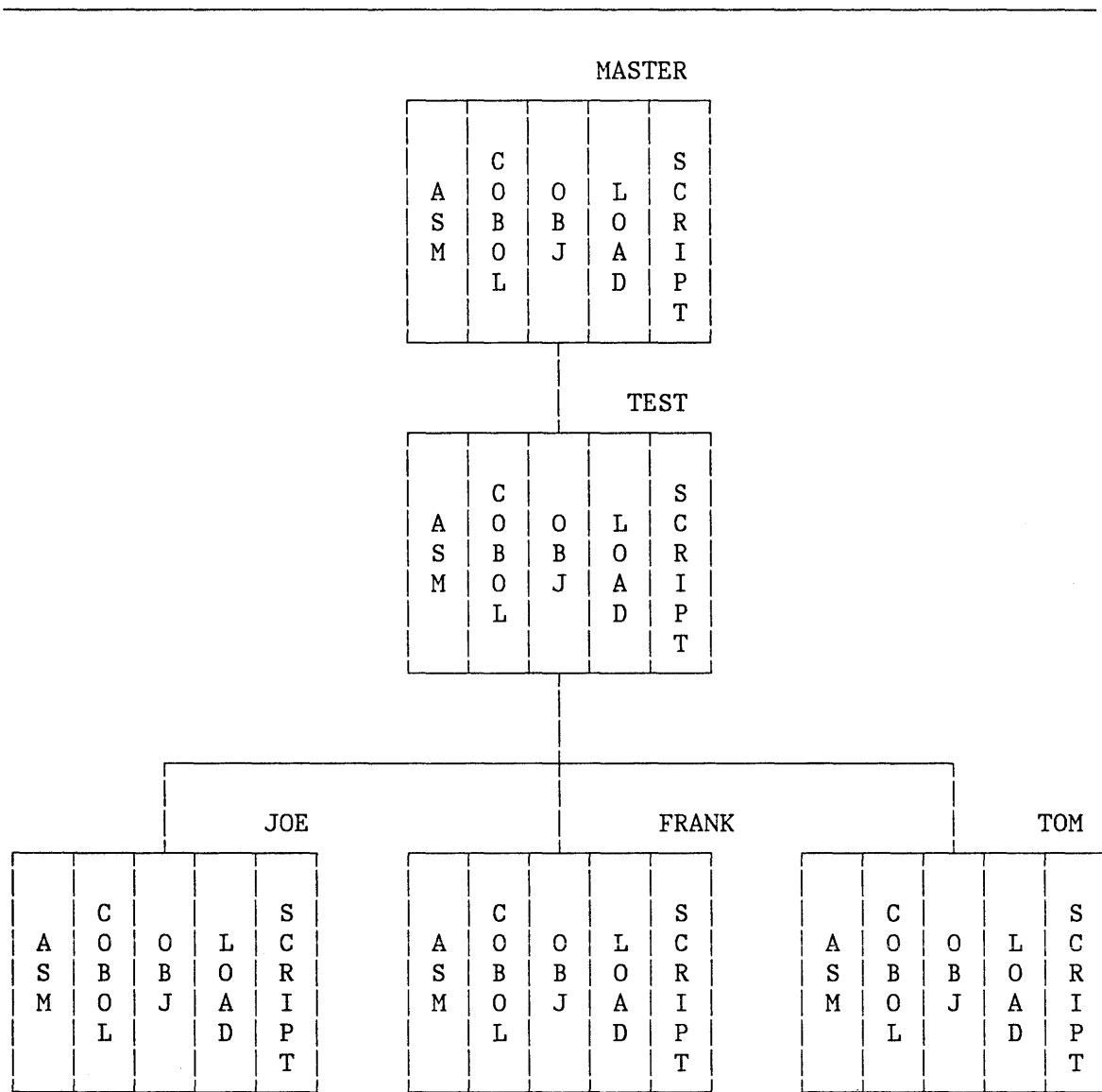


Figure 10. Hierarchy of ISPF Libraries

## 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

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

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

**Initial No. Lines:** Initial size of this version

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

**User Id:** Who created or last updated this version

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

The statistics are displayed and printed with the following formats:

- For version number and modification level: VV.MM (e.g., "LEVEL 02.15" means version 2, modification 15).
- For creation date and date last modified: YY/MM/DD (e.g., "82/06/27" means June 27, 1982).
- For time last modified: HH:MM (e.g., "17:20" means 5:20 PM).

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

On member lists, the following column headings are used:

|               |  |
|---------------|--|
| NAME          | - Name of the Member                                   |
| LIB           | - Library Number (See below)                           |
| VER.MOD       | - Version Number and Modification Level                |
| CREATED       | - Creation Date  |
| LAST MODIFIED | - Date and Time Last Modified                          |
| SIZE          | - Current Number of Lines                              |
| INIT          | - Initial Number of Lines (when member is first saved) |
| MOD           | - Number of Modified Lines                             |
| ID            | - User ID  |

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

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

```

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

```

Figure 11. Member List Display

## 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, library 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.

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

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

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.

**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 "remembers" 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), type, and password
- 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 another user (or, in split screen mode, on the other logical screen). To display an up-to-date list, return to the browse or edit entry panel, leave the member name blank 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 "snapshots". The list file accumulates, under user option, a listing of any source module that is created or modified by the PDF editor. Source listings and other types of printed output may also be obtained using the PDF utilities. A screen "snapshot" may be obtained by entering the PRINT or PRINT-HI command.

The log file maintains a log of significant user activities, including ISPF log messages, a record of 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.SPFEDITA.BACKUP
user-id.SPFEDITB.BACKUP
user-id.SPFxxx.OUTLIST
```

where "n" is the number 1 or 2, depending on which logical screen is 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 a variety of 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 12 .

---

```
----- 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 3277 terminal with 12 PF keys
```

Figure 12. Parameter Selection Panel

---

### SPECIFY TERMINAL CHARACTERISTICS (OPTION 0.1)

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

---

```

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

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

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

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

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

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

```

Figure 13. Terminal Characteristics Panel

---

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

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

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

**Note:** One or more of following installation-dependent options for terminal type may also be included on this panel:

- 3278CF - for 3278 Canadian French terminals
- 3277KN - for 3277 Katakana terminals
- 3278KN - for 3278 Katakana terminals

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

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.

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

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

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

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

**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)

When you select this option, a panel (Figure 14) 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    ==> J_                Process option    ==> J
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:
  J - Submit job to print (and delete)      K - Keep data set (do not print)
  L - Route to local printer (and delete)   D - Delete data set (do not print)

JOB STATEMENT INFORMATION: (If option "J" selected)
==> //HOSTETLA JOB (U602,B043),'HOSTETLER RS',NOTIFY=HOSTETL
==>
==>
==>

```

Figure 14. 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 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.

**Note:** 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 batch job printing, you must also specify SYSOUT class and job statement information. (If you specify option J 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.

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)

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

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

```

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

```

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

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

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

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

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

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

---

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

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

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

INSTRUCTIONS:

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

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

---



---

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

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

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

INSTRUCTIONS:

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

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

---

## CHAPTER 5. BROWSE (OPTION 1)

The browse option allows you to display source data and listings stored in ISPF libraries or 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 32,767 characters

When you select browse, a panel (Figure 17) is displayed to allow you to specify an ISPF library or other data set, volume serial (if the data set is not cataloged), and data set password (if the data set is protected).

---

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

ISPF LIBRARY:
  PROJECT ==> -
  LIBRARY ==>
  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)
```

Figure 17. Browse - Entry Panel

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 may be selected.

The browse data display is shown in Figure 18. Any invalid (nondisplayable) characters in the data are displayed as periods. Printer control characters, if present, are not displayed and are not treated as part of the data.

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

---

```
BROWSE - SPFDEMO.XXX.COBOL(CBLMAIN) - 01.01 ----- LINE 000000 COL 001 080
COMMAND ==> _                               SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. 'F20D1000'.
000300 DATE-COMPILED. OCT. 20, 1981
000400 ENVIRONMENT DIVISION.
000500 CONFIGURATION SECTION.
000600 SOURCE-COMPUTER. IBM-370.
000700 OBJECT-COMPUTER. IBM-370.
000800 INPUT-OUTPUT SECTION.
000900 FILE-CONTROL.
001000     SELECT OLD-COMREC ASSIGN TO DA-S-DD1.
001100     SELECT D1-REPORTS ASSIGN TO UR-S-D1OUT.
001200     SELECT OPTION-CARD-FILE ASSIGN TO UR-S-SYSIN.
001300 DATA DIVISION.
001400 FILE SECTION.
001500 FD  OLD-COMREC
001600     LABEL RECORD IS STANDARD
001700     RECORDING MODE IS F
001800     BLOCK CONTAINS 0 RECORDS
001900     DATA RECORD IS COMREC1.
002000 01  COMREC1.
002100     02 DUMMY           PICTURE X(520).
```

Figure 18. Browse - Data Display

---

Browse provides six functions, described in the following sections, each of which is controlled by a command that 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)
- Setting Capitalization (CAPS)
- Controlling Hexadecimal Display (HEX)

To terminate browse, enter the END command, which causes a return to the previous display (either the member list or the browse entry panel). When the return is to the member list, the member just browsed appears at the top of the list. You may select another member from the list or 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, a period followed by up to seven alphanumeric characters, 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 19. The digits on the identification line indicate the "tens" positions, e.g., "1" indicates column 10, "2" indicates column 20, etc. The plus signs (+) indicate the "fives" positions.

---

```

BROWSE - 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 19. Browse - Column Identification Line

---

## FINDING CHARACTER STRINGS

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

- Starting point and direction of search
- Special conditions to 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.

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

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

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

If you type the ALL operand, the FIND command searches for all occurrences of the string, starting at the top of the data, and positions the cursor under the first occurrence. The verification message indicates the number of occurrences found. Use the scrolling keys 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 one additional parameter that need not concern the browse user.) The full description of the FIND command is given in Chapter 6 in the section entitled "Finding and Changing Data."

## TRANSLATING CHARACTER STRINGS

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

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

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

If you omit the ON/OFF operand, ON is assumed. Under browse, caps mode is always initialized ON, except for text type data sets, which are initialized to CAPS OFF.

**Note:** The "TOP OF DATA" and "BOTTOM OF DATA" lines contain an indication of the current setting of caps mode. See Figure 18 on page 46.

*F T 'pattern'*

*ignores upper/lower-case*

*(see page 81)*

## 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 20). 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 Chapter 6.



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

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

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

The VERT (vertical) and DATA operands are valid only when hex mode is turned on. VERT causes the hexadecimal representations to be displayed vertically (two rows per byte) under each character, as shown in Figure 20. DATA causes the hexadecimal representations to be displayed as a string of hex characters (two per byte). Since the hex string is twice as long as the data string, it consumes two rows. If you omit this operand, VERT is the default. A representation of the HEX DATA display is shown in the section on the HEX command under 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'..
DCDDCECDECECD444ECC47CCDCC4DCCCC77CCCE744
95795255313965B002550A697956095694ED2857DBB
-----
```

Figure 20. Browse - Hexadecimal Display, Vertical

---

## CHAPTER 6. EDIT (OPTION 2)

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

- Invalid (nondisplayable) characters are replaced on the screen (but not in the data) with attribute bytes, which are displayed as blanks and may not be overtyped. You may display and edit invalid characters by entering hex mode (see the description of the HEX command) or by using the FIND and CHANGE commands.
- 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 21). The quote marks on the left of the screen will be filled in with sequence numbers when you enter information on the lines.



---

```

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

ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  LIBRARY ==> 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)

```

Figure 22. Edit - Entry Panel

---

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

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 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, an ABEND 80A occurs, indicating that a larger region size is required.

```

EDIT --- SPFDemo.MYLIB.PLI -----
COMMAND ==>
NAME      LIB      VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT   1         01.00  81/01/09  81/01/09 17:07    21   21   0  HUNTER
ACCT1     1         01.01  81/02/11  81/04/23 14:52   199  193   0  BECKETT
ACCT2     1         01.00  81/03/09  81/03/09 17:07    20   20   0  BECKETT
ACCT3     2         01.02  81/01/09  81/04/23 15:07    21   21   3  FISHERL
ACCT4     2         01.00  81/04/24  81/04/30 15:04    99   99   0  FISHERL
ACCT5     2         01.01  81/04/13  81/04/23 15:09    22   20   4  FISHERL
COINS     1         01.04  81/04/24  81/04/28 16:20    19   19   4  JOSLIN
COMPX     1         01.00  81/01/21  81/01/21 11:08    44   44   0  JOSLIN
COMPY     1         01.01  81/01/14  81/01/16 12:30    13   13   1  JOSLIN
DCLS     1         01.00  81/04/23  81/04/23 15:14    20   20   0  LEESB
LISTNEW   1         01.02  81/04/23  81/05/06 10:00    17   13   6  LEESB
MAIN      1         01.00  81/01/09  81/01/09 17:08     4    4   0  KIRK
MINUS     2         01.03  80/10/06  81/01/09 08:57    19   19   2  MAURER
PLUS      2         01.00  81/01/09  81/02/13 10:08    44   44   0  MAURER
TESTDIR   1         01.02  81/04/23  81/05/06 09:04    30   43  10  LAIDLAW
UPDATE    1         01.00  81/03/26  81/04/01 13:08    13   13   0  COONS
ZCOMP     2         01.01  81/01/14  81/03/13 14:30    13   13   1  STALEY
***END***

```

Figure 23. 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 72-column data field (Figure 24). The line command fields contain numbers that reflect the contents of the sequence numbers in the data. If the data has no sequence numbers, the line command fields contain numbers that start at 1 and are incremented by 1.

---

```

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

```

Figure 24. Edit - Data Display

---

## SUMMARY OF EDIT OPERATIONS

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

### Command Functions

Two types of commands are used to control edit operations:

- Primary commands, entered in the command field at the top of the screen (line 2)
- Line commands, entered in the line command field on any line by overtyping the number displayed in that field

You can use primary commands to:

- Control your editing environment

- Locate a specific line
- Find and change a character string
- 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

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

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

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

## Edit Modes

Edit can operate in the following modes:

- Number Mode - controls validation and generation of sequence numbers.
- Autonom Mode - controls automatic renumbering when the data is saved.
- Stats Mode - controls generation of library statistics.
- Autolist Mode - controls automatic recording of source listings in the ISPF list file.

- Caps Mode - controls translation of alphabetic characters to uppercase.
- Nulls Mode - controls replacement of trailing blanks with null characters on the display screen.
- Tabs Mode - controls use of "hardware" or "logical" tabs.
- Hex Mode - controls display of data in hexadecimal format.
- Recovery Mode - controls automatic recording of edit transactions to facilitate recovery following a system failure.

You can turn each mode on or off independently of the other modes with the following primary commands: NUMBER, AUTONUM, STATS, AUTOLIST, CAPS, NULLS, TABS, HEX, and RECOVERY. See the command descriptions for further explanation.

The current settings of the modes, together with the current MASK, TABS, and BOUNDS lines, are maintained in an edit profile that you can display at any time with the PROFILE command. When you terminate edit, the editor saves the current edit profile automatically as part of your user profile.

## Edit Profiles

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

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

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

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



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

- Number mode is set on or off depending upon whether the data currently contains sequence numbers.
- Caps mode is set on or off depending upon whether the data currently contains lowercase alphabetic characters.
- Stats mode is set on if the member currently has statistics. If stats mode is already on and the member has no statistics, a caution message is displayed but stats mode is left on. (Stats mode is ignored for sequential data sets.)

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

---

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

Figure 25. Edit - Caution Message

You can remove caution messages from the display by using the RESET primary command or by deleting the individual message lines using the D line command. Caution messages are never saved as part of the data.

## EDIT TERMINATION

Normally, you complete your edit session by entering the END command, which causes the following actions:

1. If any changes have been made to the data:
  - The data is renumbered if both number mode and autonum mode are on.
  - The data is automatically saved -- see the description of the SAVE command.

**Note:** "Special" lines (MASK, TABS, COLS, BOUNDS, message lines, and profile display lines) are never saved as part of the data, and need not be deleted prior to entering a SAVE command or an END command.
  - The statistics are updated (or generated if none previously existed) if stats mode is on and the data is a member of an ISPF library (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. A return is then made to the previous display (either the member list or the edit entry panel). When return is to the member list, the member just edited appears at the top of the list.

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

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

## PRIMARY COMMANDS AND FUNCTIONS

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

## Controlling Your Edit Environment

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

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

The conditions that make up your edit environment are:

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

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

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

### Sequence Number Generation and Control

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

|                |   |
|----------------|---|
| <b>NUMBER</b>  | Verifies that lines are correctly numbered, and rennumbers them to maintain the proper sequence |
| <b>RENUM</b>   | Rennumbers all lines  |
| <b>UNNUM</b>   | Turns off numbering   |
| <b>AUTONUM</b> | Automatically rennumbers 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 rennumbered 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.

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]
NUM    [OFF]
```

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

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

```
UNNUM
UNN
```

The UNNUM command has no operands.

#### AUTONUM Command

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

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

If you omit the operand, ON is assumed.

### Statistics Generation (STATS Command)

The STATS command controls the generation of statistics for a member of 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."

### 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 if you have changed the data, 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.

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

### Character Translation (CAPS Command)

The CAPS command controls the caps mode. If caps mode is on, alphabetic data that you enter at the terminal is automatically translated to

uppercase during edit operations. If caps mode is off, alphabetic data is left "as is." Caps mode is normally on for program development work and turned off when documentation (text) is being edited.

Caps mode also determines whether character strings, entered as operands of the FIND and CHANGE commands, are translated to uppercase. Note that the keyword operands and all panel parameters are always translated to uppercase regardless of the current setting of caps mode.

The CAPS command has the following format:

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

If you omit the operand, ON is assumed.

Whenever data is fetched for editing, PDF initializes the setting of caps mode based on whether or not the data contains lowercase alphabetic characters. If lowercase alphabetic characters are present in the data, caps mode is set to off; otherwise, it is set to on. 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.

#### Nulls Control (NULLS Command)

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

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

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

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

The ALL operand is valid only when nulls mode is turned on. 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.

#### Tabs Definition (TABS Primary Command)

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

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

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

The advantage of hardware tabs is that no program interrupt is generated when you press the tab forward and tab backward keys. The disadvantage is that each attribute byte occupies a screen position and may not be overtyped. The attribute bytes display as blanks.

**Logical tabs** are used by edit to break up strings of data entered on a line and reposition the strings to user-defined tab positions. The beginning of each string is indicated by a user-designated special character. 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. When you turn tabs mode off, the attribute bytes are removed and the overlaid characters at each tab position are redisplayed.

In tabs mode, you can cause the attribute bytes to be temporarily removed from a single line by blanking out the entire line command field or by placing the cursor directly under one of the attribute bytes and pressing ENTER. When you press ENTER again, the attribute bytes are reinserted.

## Profile Display and Control (PROFILE Command)

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

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

Both operands are optional, and may be entered in either order. The name operand consists of up to eight alphanumeric characters, the first of which must be alphabetic. The number operand consists of a single digit in the range 0 to 7, 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 26). In the figure, the first three lines show the current mode settings. The remaining lines show the current contents of the TABS, MASK, and BOUNDS commands, together with the COLS positioning line. The MASK and TABS commands are not displayed if they contain all blanks, and the BOUNDS command is not displayed if it contains the default boundary positions. The remainder of the figure contains a portion of the data set being edited.

If you include the name operand, and if you already have a profile of that name, edit immediately switches to the specified profile and displays it. (You can avoid displaying the profile by typing a zero for the number operand.) If a profile of the specified name does not already exist, a new profile is defined with that name. The initial content of the new profile is the same 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 6, that number of lines of the profile is displayed in the order shown in Figure 26. If you type the number 7, the complete profile is displayed, even if the mask and tabs are blanks and bounds contains the defaults. For the display in the figure, the user has entered "profile 7" to display all the lines, even the blank lines.

You can control the use of profiles from the edit entry panel. If you leave the profile name field blank, the profile name defaults to the data set type (last qualifier in the data set name). If you enter a name, it overrides the type qualifier. In either case, if a profile of that name currently exists, it is used. If it does not exist, a new profile is defined. The initial content of the new profile has the default mode settings, all-blank mask and tabs, and default bounds.

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> ....AUTONUM ON....AUTOLIST OFF....STATS ON.....
=TABS>
=MASK>
=BND>
=COLS> -----1-----2-----3-----4-----5-----6-----7--
000800          BROMLIST = OFF;          /*      TURN OFF LIST FLAG  */
000900          BROSMEMB = BROMMEMB;      /*      MOVE MEMBER NAME TO  */
001000          END;                      /*      'SELECTED' MEMBER    */
001100          IF BROSMEMB(1) ^= ' ' THEN  /*      IF MEMBER SELECTED,  */
001200          DO;                          /*
001300          %INCLUDE SYSLIB(BROFINDM);    /*      FIND MEMBER          */
001400          IF BRORCODE = 0 THEN          /*      IF NO ERRORS,        */
001500          DO;                          /*
001600          %INCLUDE SYSLIB(BROPDSH);     /*      SET UP PDS HEADER    */
001700          CALL CBRO (TLD,TDC);          /*      CALL COMMON BROWSE   */
001800          END;                          /*
001900          ELSE                            /*      ELSE,                 */
002000          DO;                          /*
002100          BROMLIST = OFF;                /*      TURN OFF LIST FLAG  */

```

Figure 26. Edit - Profile Display

### Hexadecimal Display (HEX Command)

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

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

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



## General Reset (RESET Command)

Edit displays warning messages in the data portion of the display if a mode setting has been automatically changed by edit, or if the data contains invalid characters. Edit also displays line messages to show the current setting of edit modes (when the PROFILE command is entered). You can reset the display of messages and other special conditions using the RESET command. The command has the following format:

```
RESET
RES
```

This command has no operands.

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

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

## Recovery Control (RECOVERY Command)

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

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

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

If you omit the operand, ON is assumed.

Operating with recovery mode off eliminates the I/O operations that maintain the recovery 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           *  
*****
```

Data set SPFDEMO.SAMPLE.ASM  
was being edited when a system failure or task abend occurred.

Instructions:

- Press ENTER key to continue editing the data set, or
- Enter END command to return to the primary option menu, or
- Enter CANCEL command to cancel recovery of the data set and proceed to the edit entry panel.

To continue editing a password protected data set, specify:

```
DATA SET PASSWORD ==>
```

Figure 28. Edit - Recovery Panel

---

### Locating Lines (LOCATE Command)

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

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

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

## Finding and Changing Data (FIND and CHANGE Commands)

The FIND and CHANGE commands, described in this section, allow you to find a specified character string and to change one character string to another. 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) they are discussed together. Both commands allow special forms of character strings and several optional parameters that control:

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

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

### Command Summary

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

```
FIND string-1 [ALL]
F
```

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

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

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



If the string is not found between the current cursor location and the end of data, a "BOTTOM OF DATA REACHED" message is displayed and the audible alarm (if installed) is sounded. Use the RFIND or RCHANGE PF key to wrap to the top of data and continue searching. If the string is not found anywhere in the data, a "NO string-1 FOUND" message is displayed.

If you enter the ALL operand, the FIND or CHANGE command searches for all occurrences of string-1, starting at the top of the data, and positions the cursor under the first occurrence. The 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
```

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

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

```
FIND 'every one'
CHANGE 'every one' 'all'
```

to find the next occurrence of "every one" or to change the next occurrence of "every one" to "all".

**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.

You can use the RFIND and RCHANGE PF keys together to find each occurrence of a character string, examine it, and then either change it

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

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

```
FIND ALL XYZ
```

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

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

---

```
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 29 (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 29 (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 and CHANGE, showing all optional parameters, are:

```

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

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

```

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

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

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

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

- A simple string: Any series of characters not starting or ending with an apostrophe or quotation mark, and not containing any imbedded blanks or commas.
- A delimited string: Any string starting and ending with an apostrophe (') but not containing imbedded apostrophes, or starting and ending with a quotation mark (") but not containing imbedded quotation marks.
- A hex string: Any delimited string of valid hexadecimal characters, preceded or followed by the character X. Example: X'C27B'
- A text string: Any delimited string of characters, preceded or followed by the character T. Example: T'conditions for' See the discussion under "Use of Text Strings."
- A picture string: Any delimited string of picture characters, preceded or followed by the character P. Example: P'.' See the discussion under "Use of Picture Strings."
- A single asterisk (\*). This causes the value that was used as string-1 in either a previous FIND or CHANGE command to be used again.

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

- You cannot specify a text or picture string for string-2.
- If you enter a single asterisk, the previous value of string-2 that was used in a CHANGE command is used again.

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

## Starting Point, Direction, and Extent of Search

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

|              |   |
|--------------|---|
| <b>NEXT</b>  | The scan starts at the current cursor location and proceeds forward to find the next occurrence of string-1. This is the default operand. |
| <b>ALL</b>   | The scan starts at the top of the data and proceeds forward to find all occurrences of string-1.  |
| <b>FIRST</b> | The scan starts at the top of the data and proceeds forward to find the first occurrence of string-1.                                     |
| <b>LAST</b>  | The scan starts at the bottom of the data and proceeds backwards to find the last occurrence of string-1.                                 |
| <b>PREV</b>  | 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 nonalphanumeric character (i.e., a special character or blank).

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

|             |   |           |             |            |              |               |              |                 |             |
|-------------|---|-----------|-------------|------------|--------------|---------------|--------------|-----------------|-------------|
| CHARS 'DO'  | - | <u>DO</u> | <u>DONT</u> | <u>ADO</u> | <u>ADOPT</u> | ' <u>DO</u> ' | + <u>ADO</u> | ( <u>DONT</u> ) | <u>ADO-</u> |
| PREFIX 'DO' | - | DO        | <u>DONT</u> | ADO        | ADOPT        | 'DO'          | +ADO         | ( <u>DONT</u> ) | ADO-        |
| SUFFIX 'DO' | - | DO        | DONT        | <u>ADO</u> | ADOPT        | 'DO'          | + <u>ADO</u> | (DONT)          | <u>ADO-</u> |
| WORD 'DO'   | - | <u>DO</u> | DONT        | ADO        | ADOPT        | ' <u>DO</u> ' | +ADO         | (DONT)          | ADO-        |

If no operand is specified, the default is CHARS.

### Use of Text Strings

A text (T) string, which may be used as string-1 in a FIND or CHANGE command, allows the search to be satisfied without regard to uppercase or lowercase alphabetic characters. For example, FIND T'CONDITION NO. 1' would successfully find any of the following:

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

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

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

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

### Use of Picture Strings

A picture string, which may be used as string-1 in a FIND or CHANGE command, allows the user to search for a particular type of character, without regard for the specific character involved. You can use special characters within the picture string to represent the type of character to be found, as follows:

| <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.

Examples of picture strings:

P'###' - a string of three numeric characters  
P' - ' - any two nonblank characters separated by a blank  
P'.' - any nondisplayable character  
P' #' - a blank followed by a numeric character  
P'#AB' - a numeric character followed by 'AB'

Examples of FIND and CHANGE commands using picture strings:

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

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

FIND P'..'

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

### Range and Column Limitations

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

X     Scan only lines that are excluded from the display.  
NX    Scan only lines that are not excluded from the display.

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

The col-1 and col-2 operands allow you to search only a portion of each line, rather than the complete line. These operands, integers separated by a comma or at least one blank, indicate the starting and ending

columns for the search. If you specify col-1 without col-2, the string is found only if it starts in the specified column. If you specify neither col-1 nor col-2, the search continues across all columns within the current boundary columns (for edit - see the description of the BOUNDS command), or across all columns in the data (for browse).

## Merging and Segmenting Data

You can merge or segment data by moving or copying lines into or out of the data being edited. To merge data, a primary command is used to specify the source of the move or copy, and a line command is used to specify the destination. To segment data, 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 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]
```

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

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 operand, you must specify a destination with an A or B line command. 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 30 , 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 30, 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 30 (Part 1 of 2). Edit - COPY Command Example

---

---

```

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

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

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 ENTER to copy.
Enter END command to cancel copy.

```

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

---

### Segmenting Data (CREATE and REPLACE Commands)

Use the CREATE and REPLACE primary commands to specify a 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]
CRE

REPLACE [member-name]
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 operand, you must specify the source with an

M (move) or C (copy) line command and then a panel is displayed to allow you to specify any 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 31 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 31 (Part 1 of 2). Edit - REPLACE Command Example

---

---

```
----- EDIT - REPLACE -----  
COMMAND ==>  
"CURRENT" DATA SET: SPFDEMO.XXX.COBOL(CBLCODE1)  
  
TO ISPF LIBRARY:  
PROJECT ==> SPFDEMO  
LIBRARY ==> 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 ENTER key to replace.  
Enter END command to cancel replace.
```

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

---

### Submitting to a Job Stream (SUBMIT Command)

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

```
SUBMIT  
SUB
```

This command has no operands. 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.

## 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 back 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 due to I/O errors or insufficient space, a message is displayed in the upper right corner of the screen, accompanied by an audible alarm (if installed). You can then attempt to save the data in another 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 ENTER.
3. Fill in the data set (and member) name on the CREATE or REPLACE panel, and press ENTER.

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

Alternatively, you can enter split screen mode and attempt to fix the problem by using appropriate utilities, e.g., compress (for a 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 ENTER.

When you enter a line command or modify data and press the ENTER key, edit frequently repositions the cursor to another line number or to another location within the data. Edit attempts to anticipate the most appropriate position for the cursor, based on your previous action. To assist you in locating the cursor, edit intensifies the line number field on the line that contains the cursor. Use the NEW LINE key to move the cursor to the line command area of the next line. If the cursor is in a line command area, use the TAB FORWARD key to space over to the data field.

## Rules for Entering Line Commands

The following rules apply to all edit line commands:

- You can enter several line commands, and make multiple data modifications, before you press the ENTER key. Error messages are displayed if the line commands are ambiguous. 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 ENTER, another new line is automatically inserted following that line. This allows line after line of data to be generated in a "continuous insert" mode.

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



## D (Delete) Command

To delete a line, enter the single character "D" in the line command area, overtyping the line number. A number may follow the "D" to indicate that more than one line is to be deleted. For example, "D99999" causes all remaining lines to be deleted, starting with the line containing the "D99999". The command has the following format:

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

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

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

---

```
EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- COLUMNS 001 072
COMMAND ==>                                     SCROLL ==> HALF
***** ***** TOP OF DATA *****
000100 COINS:
000200   PROCEDURE OPTIONS (MAIN);
000300     DECLARE
000400       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500       HALVES   FIXED BINARY (31),
000600       QUARTERS FIXED BINARY (31),
000700       DIMES    FIXED BINARY (31),
000800       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;
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 32 (Part 1 of 2). Edit - Insert and Delete Lines Example

---

---

```

EDIT --- SPFDemo.MYLIB.PLI(COINS) - 01.04 ----- COLUMNS 001 072
COMMAND ==>                                     SCROLL ==> HALF
***** ***** TOP OF DATA *****
000100 COINS:
000200   PROCEDURE OPTIONS (MAIN);
000300     DECLARE
000400       COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500       HALVES   FIXED BINARY (31),
000600       QUARTERS FIXED BINARY (31),
000700       DIMES    FIXED BINARY (31),
000800       NICKELS  FIXED BINARY (31),
000900       -
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     COUNT = COUNT + 1;
001600     END;
001700   END;
001800 END;
001900 END COINS;
***** ***** BOTTOM OF DATA *****

```

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

---

### R (Repeat) Command

To repeat a line, enter the single character "R" in the line command area, overtyping the line number. When you press ENTER, the editor inserts a duplicate copy of the line immediately following the line containing the "R". A number may follow the "R" to indicate that the line is to be repeated more than one time. For example, "R5" would cause five identical copies of the line to be inserted. The command has the following format:

```

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

```

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

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

As an example of the use of the R 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 33.

---

```

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

```

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

---

```

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

```

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

---

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

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

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

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

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

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

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

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

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

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

---

```

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

```

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

---

---

```

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

```

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

---

## Preparing Text

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

```

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

```

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

```

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

```

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

### TE (Text Entry) Command

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

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

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

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

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

You can enter multiple paragraphs in either of two ways:

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

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

## TS (Text Split) Command

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

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

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

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

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 35 (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. _
000061 !!!!!
000062 !!!!!
000063 !!!!!
000064 To split a line, enter the characters "TS" in a line
000065 command area and then move the cursor to the desired split point (on the
000066 same line) before pressing the ENTER key. A new line is inserted
000067 following the line containing the "TS", and all text to the right of the
000068 cursor is moved to the beginning of the next line (following the
000069 inserted line). A number may follow the "TS" to cause additional lines
000070 to be inserted.
000071
000072 .FO OFF
000073 .BX 9 72
000074     TS   - Text split and insert line
000075     TSn  - Text split and insert "n" lines
000076 .BX OFF
000077 .CO ON
```

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

---

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

---

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

---

### TF (Text Flow) Command

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

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

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

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

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

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

---

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

---

## Overlaying Lines (O Command)

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

Enter the character "O" in the line command area to specify the line over which data is to be moved or copied. Enter a number following the "O" to indicate the number of lines to be 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.

A before-and-after example of overlaying lines is shown in Figure 38. Generally, a list such as that shown in the figure would be maintained and edited as a single column (left adjusted). Prior to the overlay, portions of the list would be right-shifted by the appropriate amounts to overlay in the multicolumn format.

---

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

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

---

---

```

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

```

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

---

### Identifying Columns (COLS Command)

You can display a 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 39. 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 39. 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. Whenever a new line is inserted, it is pre-filled with the contents of the mask. Initially, the mask contains all blanks. You can define the mask by entering characters when the blank mask is displayed.

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

MASK - Display mask definition line

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



example, the mask has been displayed and the characters "/" and "/" have been entered at the desired positions on the mask line. Five new lines are then inserted following line 1700. The second part of the example shows that the new lines contain the contents of the mask.

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

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

---

```

EDIT --- SPFDEMO.SAMPLE.PLI(ED@21PM) - 01.01 ----- COLUMNS 001 072
COMMAND ==>                                     SCROLL ==> HALF
001100      ELSE
001200 /*                COMPUTE PARM LENGTHS          */
001300      DO;
=COLS>  -----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----
=MASK>                                     /*                */
001400      LP = 8;
001500      DO WHILE (ED@MPROJ(LP) = ' ');
001600      LP = LP-1;
I5      END;
001800      DO K=1 TO 4;
001900      IF ED@MLIB(K,1) = ' ' THEN
002000      DO;
002100      LL(K) = 8;
002200      DO WHILE
002300      (ED@MLIB(K,LL(K)) = ' ');
002400      LL(K) = LL(K)-1;
002500      END;
002600      END;
002700      ELSE
002800      LL(K) = 0;
002900      END;
***** ***** BOTTOM OF DATA *****

```

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

---

```

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

```

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

## Controlling Tabs (TABS Line Command)

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

```

TABS - Display tab definition line
TAB

```

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

To define hardware and logical tab positions, type an asterisk (\*) at each desired column position on the tab definition line. The asterisks

indicate the locations at which the attribute bytes will be inserted (for hardware tabs) or the locations following which strings will be repositioned (for logical tabs). For example, if your tab character is \$ and you enter:

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

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

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

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

Figure 41 shows an example of the tab definition line displayed together with the column identification line. The tab line shows a software tab field extending from columns 2 through 39, and a hardware/logical tab definition at column 43.

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

The tab definitions remain in effect (even if not displayed) until you change them. Tab definitions are retained in the current edit profile, and are automatically used the next time you edit the same type of data.

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

```

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

```

Figure 41. Edit - Tab Definition Line

## Defining Boundary Columns (BOUNDS Command)

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

```

BOUNDS - Display boundary definition line
BOUND
BNDS
BND

```

The column boundaries are used to limit the scope of:

- Left- and right-shift line commands
- FIND and CHANGE commands when explicit columns are not specified
- TE (text entry), TS (text split), and TF (text flow) line commands

- Overlay line command
- Left and right scrolling

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

The boundaries are initialized by edit based on the data 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 42.

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

Figure 42. Edit - Default Column Boundary Settings

You can change the boundaries by displaying the boundary definition line and changing its contents. The boundary definition line contains a single "<" character indicating the position of the left column boundary, and a single ">" character indicating the position of the right column boundary. If you have already changed the boundaries, you can revert to the default bounds by displaying the boundary definition line and blanking out its contents with the ERASE EOF key.

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

Figure 43 shows an example of the boundary definition line displayed together with the column identification line. In this example, the left bound is currently set at column 42 and the right bound at column 70.

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

---

```

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

```

Figure 43. Edit - Boundary Definition Line

---

## Changing Indentations

Use the shift line commands to shift the contents of a line (or block of lines) left or right. The forms of these commands are:

```

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

```

Data shifting attempts to shift the body of a program statement without shifting the label or comments, and prevents loss of data. Column shifting shifts information within column boundaries without regard to its contents, and may result in the loss of data.

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

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

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

To shift a block of lines left, enter "<<" or "(" in the line command area of the first and last lines to be shifted. The first and last lines need not be on the same page; you can scroll between entering the first command and the second command.

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

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

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

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

Shifting occurs within column boundaries that you can change. Default column boundaries are established by edit, based on the data 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.

### > 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.

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

---



---

```

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

```

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

---

#### ) and ( (Column Shifting) Commands

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

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

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

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

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

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

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

Figure 45 shows a before-and-after example of excluded lines. The block of lines, 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 45 (Part 1 of 2). Edit - Excluded Lines Example

---

```

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

```

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

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

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

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

Figure 46 shows a before-and-after example of redisplaying excluded lines. The command "S3" is entered in the line command area of the "EXCLUDED LINES" message. Three lines with the highest indentation level are redisplayed.

You can enter any line command that normally operates on a single line in the line command area of the "EXCLUDED LINES" message. For example, a "D" entered on that line deletes the entire block of excluded lines.

This feature is particularly useful in conjunction with the shift commands. Suppose, for example, you want to "data" shift all lines to the left by four column positions. Proceed as follows:

1. On the first data line, enter "X99999". This excludes all lines.
2. On the "EXCLUDED LINES" message, enter the shift command "<4".
3. In the primary command area, enter the RESET command to bring the lines back into view.

**Note:** Excluded lines need not be redisplayed before saving the data. The "EXCLUDED LINES" message line is never saved 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 = BROSMEMB;          /*      MOVE MEMBER NAME TO     */
001300      END;                          /*      'SELECTED' MEMBER      */
001400      IF BROSMEMB(1) = ' ' THEN     /*      IF MEMBER SELECTED,    */
001500      DO;                            /*                                */
S3 - - - - - - - - - - - - - - - - - - - - - - - 11 LINE(S) NOT DISPLAYED
002700      END;                          /*                                */
002800      ELSE                          /*      ELSE (NOTHING SELECTED),*/
002900      BROMLIST = OFF;                /*      TURN OFF LIST FLAG      */
003000      END;                          /*      END OF LOOP            */
003100      BROSMEMB = BLANKS;             /*      CLEAR SELECTED MEMBER   */
003200      END;                          /*                                */
***** ***** BOTTOM OF DATA *****

```

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



These comments allow you to develop error handling logic for your function. They may also provide parameter descriptions for other types of models.

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

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

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

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

The first part of the logical name is the model **class**.

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

You can default this part of the logical name whenever the type qualifier of the 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.

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

---

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

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

Enter END command to cancel MODEL command.

Figure 47. Edit - Model Class Selection Panel

---

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

The command has the following format:

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

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

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

## Using Models

Models are used when you are editing a 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, use the A (after) or B (before) line command to indicate where the model will go. You need not use the A or B command when you have a new 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 overwrite 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 48).

```

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

TABLES (General)          TABLES (Row)          VARIABLES
G1  TBCREATE             R1  TBADD              V1  VGET
G2  TBOPEN               R2  TBDELETE           V2  VPUT
G3  TBQUERY              R3  TBGET
G4  TBSAVE               R4  TBPUT
G5  TBCLOSE              R5  TBMOD
G6  TBEND                R6  TBEXIST
G7  TBERASE              R7  TBSARG
                               R8  TBSCAN
                               R9  TBTOP
                               R10 TBBOTTOM
                               R11 TBSKIP
                               R12 TBVCLEAR

FILE TAILORING           DISPLAY                MISCELLANEOUS
F1  FTOPEN               D1  DISPLAY            M1  SELECT
F2  FTINCL               D2  TBDISPL           M2  CONTROL
F3  FTCLOSE              D3  SETMSG             M3  BROWSE
F4  FTERASE


```

Enter END command to cancel MODEL command.

Figure 48. Edit - Model Selection Panel for Command Models

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

You can then continue editing the 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 command to eliminate the notes from the display, leaving the service call, the error-handling logic, and the comments.

```

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

```

Figure 49. Edit - Sample Dialog Development Model

### Adding, Changing, and Deleting Models

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

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

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



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

---

```
----- PANEL MODELS -----  
OPTION ==>  
  
FORMATS:  
F1 ENTRY      - Data entry panel  
F2 MULTIPLE   - Multiple column panels  
F3 SELECTION  - Menu selection panel  
F4 TBDISPL    - Table display panel  
F5 TUTORIAL   - Help/tutorial panel  
  
STATEMENTS:  
S1 ASSIGN     - Assignment statement  
S2 ATTR       - Attribute section header - )ATTR  
S3 ATTRIB     - New attribute character definition  
S4 BODY       - Body section header - )BODY  
S5 CONTROL    - Control variables  
S6 IF         - If statement  
S7 MODEL      - Model section header - )MODEL  
S8 VER        - Verify statement  
S9 VPUT       - Variable put statement
```

Enter END command to cancel MODEL command.

Figure 51. Edit - Panel Model Selection Panel

---

---

----- PANEL MODELS -----

OPTION ==>

FORMATS:

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

STATEMENTS:

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

Enter END command to cancel MODEL command.

Figure 52. Edit - Modified Visible Panel Model Selection Panel

---

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

---

```

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

```

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

---

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

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

```

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

```

Figure 54. 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 54 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 53, for example, the name of the model for a selection panel format is ISREMMF3.

To determine the name of the model selection panel so that you can look at it to find the model member name, use the PANELID command when that panel is displayed on the screen. Then use browse or edit to look at the member of the panel library with that name.

## Changing Models

To modify a model that currently exists, copy the existing model from the skeleton data set into your own data set. Then use edit to change the model as you would change any text 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 49).

## Deleting Models

You can delete models by deleting the references to them. To delete the references, remove the entry referencing the model in both the )BODY and )PROC sections of the model selection panel.

Generally, you can leave the model itself in the skeleton library. However, if you are deleting a substantial number of models, you may wish to consider deleting those members from the library and compressing the library.

## CHAPTER 7. UTILITIES (OPTION 3)

The utility option provides a variety of functions for library, 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; and converting old format panels and messages.

The utility selection panel is shown in Figure 55.

---

```
----- 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 or copy members or data sets
 4 CATALOG   - Catalog management:
                Display or print catalog entries
                Initialize or delete user catalog alias
 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
```

Figure 55. 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 56 shows a panel on which data set name SPFDEMO.XXX.COBOL has been entered. Possible actions are:

|       |                         |
|-------|-------------------------|
| C     | - Compress data set     |
| X     | - Print index listing   |
| L     | - Print entire data set |
| P     | - Print member          |
| R     | - Rename member         |
| D     | - Delete member         |
| B     | - Browse member         |
| blank | - Display member list   |

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.

---

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

C - Compress data set          P - Print member
X - Print index listing        R - Rename member
L - Print entire data set      D - Delete member
blank - Display member list    B - Browse member

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

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

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

Figure 56. Library Utility Panel

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 a special installation procedure. It may not be available at your installation.

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 A for a discussion of the index listing format.

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 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.

Figure 57 shows a before-and-after example where members ACCT1 and ACCT2 are printed, member UPDATE is deleted, and member LISTNEW is renamed LISTOUT.

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

```

```

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

```

Figure 57. 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 58). 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

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.

---

```
----- 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
blank - Display data set information

ISPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> 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 58. Data Set Utility Panel

---

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), data set characteristics, and space allocation parameters (Figure 59). Default values are pre-entered 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)
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
```

Figure 59. 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 60). 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, the data set is not recataloged to the new data set name. 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 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.

If you specify a volume serial for the data set to be deleted, the data set is scratched but not uncataloged (PDF assumes the data set is already uncataloged). If no volume serial is specified, the data set is scratched and uncataloged.

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 volume serial, data set characteristics, current space allocation, and current space utilization are displayed (Figure 61). To return to the previous panel, press the ENTER key or enter the END command.



---

```
----- 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 ==>  
  LIBRARY ==>  
  TYPE    ==>  
  
OTHER PARTITIONED OR SEQUENTIAL DATA SET:  
  DATA SET NAME ==> 'NEWPROJ.DEVEL.COBOL' _
```

Figure 60. Data Set Utility - Rename Data Set

---

---

```

----- 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:         80/05/31

                                CURRENT UTILIZATION:
                                Used Tracks:           5
                                Used Extents:           1
                                Used Dir. Blocks:       2
                                Number of Members:      10

```

Figure 61. 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 62). Possible actions are:

```

CP - Copy data set or member(s) and print
MP - Move data set or member(s) and print
C  - Copy without printing
M  - Move without printing

```

When you press ENTER, a second panel is displayed that allows you to specify the "to" data set (Figure 62). 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.

You may specify a DASD-resident sequential or partitioned data set for either the "from" or "to" data set. The allowable data set characteristics are the same as for browse. You can move or copy load modules (stored in partitioned data sets with "undefined" record format), but you cannot print them.

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, 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) or CP (copy 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.

---

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

CP - Copy data set or member(s) and print      C - Copy without print
MP - Move data set or member(s) and print      M - Move without print

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

FROM ISPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> 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 62 (Part 1 of 2). Move/Copy Utility Panels

---

```

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

SPECIFY "TO" DATA SET BELOW

TO ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  LIBRARY ==> 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)

```

Figure 62 (Part 2 of 2). Move/Copy Utility Panels

---

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 or copied.
- Leaving the "from" member name blank causes a member list to be displayed (after the second panel has been displayed).
- Specifying a "from" member name of asterisk (\*) causes all members to be moved or copied.

If the "to" 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 or copied 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 or copied by entering an S in front of one or more member names. The options for move or copy, and print or no print must have been previously specified on the first move/copy panel. If the "to" 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.

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 63 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 ==>
          NAME      RENAME  VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT  MOD  ID
S ACCOUNT
S ACCT1
S ACCT2
  COINS
  COMPX
S COMPY      ZCOMP_
  DCLS
  LISTOUT
  MAIN
  TESTDIR
  **END**
          SCROLL ==> PAGE

```

Figure 63 (Part 1 of 2). Move/Copy Utility - Copy Members from List

---

---

```

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

```

Figure 63 (Part 2 of 2). Move/Copy Utility - Copy Members from List

---

For any move/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).

- 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 to accommodate different record lengths.
- The data to be moved or copied is not renumbered or modified in any way, except for possible truncation or padding as noted above. Printer control characters, if present, are treated as part of the data.
- 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 or copied.

For a move/copy 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 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, 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).

## CATALOG MANAGEMENT UTILITY (OPTION 3.4)

When you select this option, a panel is displayed that allows you to specify the project name for an ISPF library (or high-level qualifier for any sequential or partitioned data set), and an action to be performed (Figure 64). Possible actions are:

- I        - Initialize user catalog alias
- D        - Delete catalog alias
- P        - Print catalog entries
- blank   - Display catalog entries

If you select option I (initialize user catalog alias), you must also specify a catalog name. This action is valid only if there are no data sets currently allocated with this project name. It causes all subsequent catalog entries for this project name to be placed in the user catalog, pointed to by a single entry in the master catalog. This is accomplished by defining the project name as an alias of the designated user catalog.

**Note:** You must have already allocated space for the user catalog on the designated volume.

If you select option D (delete catalog alias), the alias for the designated project name is deleted from the master catalog.

If you select option P (print catalog entries), a catalog listing for the designated project name is recorded in the ISPF list file. The listing is produced by the Access Method Services (AMS) program. You may specify AMS list catalog parameters to control the amount of output produced, and more than one level of qualifier may be specified for project.

If you leave the option field blank (to display catalog entries), a catalog listing (produced by AMS) is displayed at the terminal in browse mode. You can scroll the listing using the scrolling commands or the LOCATE command. You may enter all browse commands from this display. To terminate the display of the listing, enter the END command, which causes a return to the catalog management utility panel.



---

```
----- CATALOG MANAGEMENT -----
OPTION ==> _

    I - Initialize (define) user catalog alias
    D - Delete user catalog alias
    P - Print catalog entries
    blank - Display catalog entries

PROJECT NAME:
    ==> SPFDEVEL

AMS LIST CATALOG PARAMETER: (If option "P" or "BLANK" selected)
    ==> NAME

CATALOG NAME: (If option "I" selected)
    ==>
```

Figure 64. Catalog Management Utility Panel

---

### 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 65). The reset utility handles only partitioned data sets with fixed or 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 three valid options for this utility:

- R - Create/update statistics, conditionally reset sequence numbers
- N - Create/update statistics, no reset of sequence numbers
- D - Delete statistics

Use the R option either to create statistics in a library that does not currently have them, or to update statistics in a library.

---

```
----- RESET ISPF STATISTICS -----
OPTION ==> R

R - Create/update ISPF statistics, conditionally reset sequence numbers
N - Create/update ISPF statistics, no reset or sequence numbers
D - Delete ISPF statistics

NEW VERSION NUMBER ==> 2          (Required for option R)
NEW TSO USERID     ==> XXXXXXXX (If user id is to be changed)

ISPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> 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 65. Reset Statistics Utility Panel

---

The data is scanned to determine if valid, ascending sequence numbers are present in all records. If so, the data is renumbered and the modification level flags (the last two digits of each sequence number) are set to zeros. If valid sequence numbers are not present, renumbering is not done.

Use the N option to create or update statistics as in option R; however, the data is not renumbered. Use this option if the data already contains statistics and you want to update the user id and version information without renumbering the data.

Use the D option to delete statistics.

The version number field is required for option R, optional for option N, and ignored for option D. If you select option N and select a member without statistics, the statistics are created as if the version had been specified as "1". If you specify a version number, the statistics are created or reset as follows:

Version number: set to the specified value  
Modification level: set to zero  
Creation date: set to current date  
Date/time last modified: set to current date and time  
Current number of lines: set to the current number of lines  
Initial number lines: set to the current number of lines  
Number of modified lines: set to zero

The user id field is optional for options R or N; it is ignored for option D. If you specify a user id, it is placed in the user id field of the statistics. If you leave the user id field blank and select a member without statistics, the user id field of the statistics is set to the current user id.

The rules for specifying member names are as follows:

- Specifying a valid member name causes the statistics to be created, updated, or deleted for a single member.
- Leaving the member name blank causes a member listing to be displayed.
- Specifying a member name of asterisk (\*) causes the statistics to be created, updated, or deleted for all 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 66). 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:

- J - Generate JCL to print or punch data set
- L - Route data set to local printer

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.

---

```

----- HARDCOPY UTILITY -----
OPTION ==> J

  J - Generate JCL to print or punch data set
  L - Route data set to local printer

DATA SET NAME ==> COMPA.LIST
DISPOSITION      ==> DELETE      (KEEP or DELETE)
VOLUME SERIAL    ==>              (If not cataloged)
DATA SET PASSWORD ==>              (If password protected)

SYSOUT CLASS     ==> A            (If option "J" selected)
LOCAL PRINTER ID ==>              (If option "L" selected)

JOB STATEMENT INFORMATION: (If option "J" selected, verify before proceeding)
==> //EKMAN2  JOB (U602,B043),'EKMAN RW',NOTIFY=EKMAN
==>
==>
==>

```

Figure 66. Hardcopy Utility Panel

---

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).

If you select option J (generate JCL), you must also specify the SYSOUT class and job statement parameters. The SYSOUT class may specify a printer or punch. 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). You may then select another data set to cause another job step to be generated.

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 overtype them, since the JOB statement has already been generated.

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.

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 select option L (route data set to local printer), you must specify a printer id. When you have entered the desired information and press the ENTER key, 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 use the L option.

## 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 67). 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.

For either option, you can limit the VTOC listing to data sets that begin with a designated high-level qualifier (specified via the project name on the list VTOC selection panel).

---

```
----- VTOC UTILITY -----
OPTION ==>

P - Print VTOC entries
blank - Display VTOC entries

VOLUME ==> 123456_
PROJECT ==>          (Blank for entire volume)
```

Figure 67. List VTOC Utility 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 68.

```

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:      7809      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 68. Display VTOC Example

## 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 69). 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 69. 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 70) 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,B043),MHK
==>
==>
==>
```

Figure 70. 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 71) is displayed to prompt you for an application id. The name of the command table is then derived by appending "CMDS" to the application id. If the table exists in the table input library, it is displayed and may be modified. If the table does not exist in the table input library, a new table is generated.

**Note:** You cannot use this utility to modify a command table that is currently in use. Command table ISPCMDS (the system command table) is always in use by the dialog manager. To modify the system command table, make a copy of the table, rename the copy, modify the copy, and then replace the original with the copy outside the ISPF environment. If you use the TSO COPY command, specify the NONUM parameter. Otherwise, you will have an unusable command table.

The named table is displayed on an edit panel (Figure 72). 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.

---

```
----- 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 71. 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 72. Command Table Editing Panel

---

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 73) 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 74) allows you to specify the "to" library, which must also be a partitioned data set.

Old-format members are read from the first ("from") library, converted to the new format, and stored in the second ("to") library. The first library is not altered by this utility.

This utility is not capable of fully converting all old-format Structured Programming Facility (SPF) menus to ISPF panels. In general, this utility:

- Handles conversion of attribute characters, including those specified with the old-format <FIELDS> statement, provided the <FIELDS> statement is the first line encountered in the old-format menu.

- Generates an )ATTR section, if required, and a )BODY section. The name of the first input field on line 2 (if any) is changed to ZCMD. Other input and output fields retain the same names as in the old-format menu.
- Generates an )INIT section with .HELP set to the name specified using the old GENHELP keyword, and with variables initialized as specified using the old INIT keywords.

The convert utility does not handle explicit cursor placement, the old KEY keyword, or the old LIST/RETURN keywords except as described below.

The convert utility fully converts:

- Old-format selection menus below the primary option level, that were designed to be processed by the SPFUTIL program in the Structured Programming Facility.

```
----- CONVERT MENU/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
LIBRARY ==> 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 73. "From" Panel for Converting Menu/Message Definitions

- Old-format tutorial pages, except that the utility cannot handle the bypassing of a tutorial page that is displayed only if explicitly selected (bypassed in the normal flow when you keep pressing the ENTER key). The converted page is not bypassed in the normal flow.

To correct the problem, you must manually change the (new-format) parent panel by inserting an asterisk in front of the panel name in the TRANS statement. For more information, see "Help/Tutorial Panels" in ISPF Dialog Management Services.

For messages, the utility properly handles the text of both short and long messages, the HELP reference (if any), and the ALARM indicator. Variable fields in old-format messages are converted to dummy variable names, beginning with an ampersand. You must change these manually to the appropriate dialog variable names.

Only new-format messages may be displayed on new-format panels, and only old-format messages may be displayed on old-format panels. The ISPF LOG service writes only new-format messages to the ISPF log file.

---

COPY --- OLD FORMAT SPF22.OURMODS.MENUS  
COMMAND ==>

SPECIFY "NEW FORMAT" DATA SET BELOW.

TO ISPF LIBRARY:  
PROJECT ==> ISP  
LIBRARY ==> 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 74. "To" Panel for Converting Menu/Message Definitions

---



## CHAPTER 8. FOREGROUND (OPTION 4)

The foreground option provides an interface to standard language processors for foreground compilation, assembly, linkage edit, or 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.

The foreground selection panel is shown in Figure 75.

When the Session Manager Program Product, 5740-XE2, is installed, a line is added to the panel from which you may enter Session Manager mode when invoking any of the foreground processors.

---

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

1 - System assembler           9 - SCRIPT/VS
2 - OS/VS COBOL compiler       10 - COBOL interactive debug
3 - FORTRAN IV (G1) compiler   11 - FORTRAN interactive debug
4 - PL/I checkout compiler
5 - PL/I optimizing compiler
6 - PASCAL/VS compiler
7 - Linkage editor
```

Figure 75. 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 76 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
LIBRARY ==> 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 76. 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 77). 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 76 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 (HX01,B001,123456),M.KIRK
==> //
==>
==>

```

Figure 77. 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/VS:   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 78). 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  
  LIBRARY ==> 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 78. Foreground - Linkage Editor Example

## SCRIPT/VS PROCESSING

When this option is selected, a sequence of panels is displayed to allow you to format, display, and print text maintained in PDF libraries or other partitioned or sequential data sets. Use of this facility requires the installation of the Document Composition Facility (SCRIPT/VS) Program Product, with the Foreground Environment Feature.

This facility has the following options:

- 1 - Draft document options
- 2 - Final document options

For the draft document options, it is assumed that your primary interest is to assure that the content, syntax, and spelling within the document are correct. The final document options provide additional flexibility and control for specifying document formats.

The first panel displayed by the SCRIPT/VS option allows you to select the draft or final options, and to specify the data set to be formatted (Figure 79). For partitioned data sets, a member list is displayed if a member name is not specified. You may specify a concatenated sequence of ISPF libraries to allow inclusion of imbedded or appended members from more than one library.

The formatted listing is stored in a data set named:

prefix.list-id.LIST

where "list-id" may be specified on the panel. If you leave this 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.

---

```

----- SCRIPT/VS SELECTION MENU -----
OPTION ==> 1_

      1 - Draft document options          2 - Final document options

ISPF LIBRARY:
PROJECT ==> SPFDEVEL
LIBRARY ==> PRM           ==>           ==>
TYPE    ==> TEXT
MEMBER  ==>                (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>

LIST ID    ==>                (Blank defaults to member name)

DATA SET PASSWORD ==>        (If password protected)

```

Figure 79. SCRIPT/VS Utility - First Panel

---

### Draft Document Options

When you select option 1 on the first SCRIPT/VS panel, a secondary panel (Figure 80) is displayed to allow you to specify the draft document options. All the parameters on this panel are optional.

For automatic spelling verification, enter "yes" in the spelling check field. You must also enable spelling verification within the document. (See the .sv control word in the Document Composition Facility User's Guide, SH20-9161.)

For all-uppercase printing, enter "yes" in the upper case only field.

For line numbers to be printed in the output document, enter "yes" in the line number field. Line numbers can be useful for correcting errors and misspelled words, which SCRIPT/VS identifies by line number.

Use the "other SCRIPT parms" field to enter any SCRIPT/VS parameters not on the panel. If use of a profile is to be inhibited, you must enter NOPROF in this field.



If a SCRIPT/VS document profile is to be used, you must enter the name of the sequential or partitioned data set (and member) that contains the profile in the profile data set field. If you leave this field blank, the default SCRIPT/VS profile is used unless "NOPROF" was entered in the other SCRIPT parms field.

To include a user GML library for formatting the data, you must enter its name in the SCRIPT lib data set field. If you specify no data set, the default GML library, if one is defined, is used.

---

```
----- SCRIPT/VS DRAFT OPTIONS -----  
COMMAND ==>  
  
DATA SET NAME: SPFDEVEL.PRM.TEXT(TOP) - 01.24  
  
SCRIPT/VS DRAFT OPTIONS:  
  SPELLING CHECK      ==> YES  (YES or NO)  
  UPPER CASE ONLY    ==> NO   (YES or NO)  
  LINE NUMBER        ==> NO   (YES or NO)  
  
OTHER SCRIPT PARS    ==> UNFORMAT  
  
PROFILE DATA SET    ==>  
  
SCRIPT "LIB" DATA SET ==>
```

Figure 80. SCRIPT/VS Draft Document Options

---

## Final Document Options

When you select option 2 on the first SCRIPT/VS panel, a secondary panel (Figure 81) is displayed to allow you to specify the final document options. All the parameters on this panel are optional. However, if you enter a device type, you must fill in the lines per inch and form definition fields correctly.

For automatic spelling verification, enter "yes" in the spelling check field. You must also enable spelling verification within the document. (See the .sv control word in the Document Composition Facility User's Guide, SH20-9161.)

For all-uppercase printing, enter "yes" in the upper case only field.

The next four fields on the panel are used to describe the output device and the physical characteristics of the printed page. Specify the type of printer on which the output will be printed (and for which SCRIPT/VS will format the document) in the device type field.

---

```
----- SCRIPT/VS FINAL OPTIONS -----  
COMMAND ==>  
  
DATA SET NAME: SPFDEVEL.PRM.TEXT(TOP) - 01.24  
  
SCRIPT/VS FINAL OPTIONS:  
  SPELLING CHECK      ==> YES      (YES or NO)  
  UPPER CASE ONLY    ==> NO       (YES or NO)  
  DEVICE TYPE        ==> 3800     (1403 OR 3800)  
  LINES/INCH         ==> 8       (6, 8, or 12)  
  PAGE WIDTH         ==> 13.5    (8.5, 11, or 13.5)  
  PAGE LENGTH        ==> 11      (8.5 or 11)  
  3800 "CHARS" PARM  ==> GT12,GT15  
  
BINDING:   ODD PAGE  ==> 10      (Specify margin)  
           EVEN PAGE ==> 10      (Specify margin)  
  
OTHER SCRIPT PARMS ==> SY (C 1 D YES) TWOPASS  
  
PROFILE DATA SET  ==> 'XYZ.MASTER.PROFILE'  
  
SCRIPT "LIB" DATA SET ==> 'XYZ.MASTER.GML'
```

Figure 81. SCRIPT/VS Final Document Options

If the document is to be printed on a device other than an IBM 1403 or 3800, leave this field blank, and specify the printer in the other SCRIPT parms field (see below).

The lines/inch, form width and form length fields are used to describe how the document should fit on the printed page.

The 3800 chars parm field allows you to specify the names of one or more type fonts if the document is being formatted for an IBM 3800 printer. You may enter the names of one or two SCRIPT/VS fonts or of up to four 3800 character arrangement tables, separated by commas. You must specify these same parameters on the foreground print panel with a JCL output/format statement.

You may specify the page margin to be used in binding the document by entering the size of the margin in the binding fields. You can specify a different margin for even and odd numbered pages. For more information, see "BIND" in Document Composition Facility User's Guide, SH20-9161.

The other SCRIPT parms field allows you to specify any SCRIPT/VS parameters not on the panel. To inhibit use of a profile, you must enter NOPROF in this field.

If you use the other SCRIPT parms field to specify a device other than the 1403 or 3800, you must have previously added the device to the SCRIPT/VS device characteristics table. You may select a device from this table by leaving the device type field blank and specifying the SCRIPT/VS device parameter in the other SCRIPT parms field.

If a SCRIPT/VS document profile is to be used, you must enter the name of the sequential or partitioned data set (and member) that contains the profile in the profile data set field. If you leave this field blank, the default SCRIPT/VS profile is used unless you entered "NOPROF" in the other SCRIPT parms field.

To include a user GML library for formatting the data, enter its name in the SCRIPT lib data set field. If no data set is specified, the default GML library, if one is defined, is used.

## CHAPTER 9. BATCH (OPTION 5)

The batch option provides an interface with standard language processors for background compilation, assembly, or linkage edit of programs stored in ISPF libraries or other partitioned or sequential data sets.

The batch selection panel is shown in Figure 82. It allows you to enter job statement information and to select a language processor.

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
    2 - OS/VS COBOL compiler
    3 - FORTRAN IV (G1) compiler
    4 - PL/I checkout compiler
    5 - PL/I optimizing compiler
    6 - PASCAL/VS compiler
    7 - Linkage editor

SOURCE DATA ONLINE ==> YES (YES or NO)

JOB STATEMENT INFORMATION: (Verify before proceeding)
==> //SILVA7   JOB (U602,B043),'SILVA RJ',CLASS=B,
==> //          MSGLEVEL=2,
==> //          NOTIFY=SILVA
==>
```

Figure 82. Batch Selection Panel

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 83.

---

```
----- BATCH PL/I OPTIMIZING COMPILE -----
COMMAND ==>

ISPF LIBRARY:
  PROJECT ==> SPFDEMO
  LIBRARY ==> 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 83. 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 83) 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=OURSYS.PLIMACS,DISP=SHR
//SYSLIN    DD   DSN=SPFDEMO.MYLIB.OBJ(TOPSEG),DISP=OLD

```

The first step executes the ISRSCAN program (distributed as part of PDF), which searches the user-specified sequence of concatenated libraries to find the designated member. If ISRSCAN 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, ISRSCAN exits with one of the following return codes, which prevent execution of the second job step:

```

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

```

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.

## 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 blank panel is displayed with the following input field:

```
ENTER TSO COMMAND OR CLIST BELOW:  
====>
```

**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.

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
```

Do not enter the following commands under PDF:

- LOGON, LOGOFF
- ISPF, PDF, and ISPSTART
- TEST
- Commands that are restricted by TSO or PCF
- Commands that require APF authorization (such as RACF commands)
- 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.
- CLIST attention exits produce unpredictable results when you press PA1.
- TERMIN command procedure statements are not recognized.

If your installation has the Session Manager Program Product installed, the panel includes an entry line:

```
ENTER SESSION MANAGER MODE ==>          (YES or NO)
```

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.

## 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 84) 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 84. 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 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<br>PRIMARY<br>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<br>LINE<br>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.

|                                 |  |
|---------------------------------|--|
| <b>CANCEL (CAN)</b>             | Ends the current option. Any changes made to the data are ignored.   |
| <b>END</b>                      | Ends the current option. Any changes made to the data now take effect.   |
| <b>LOCATE (LOC, L) argument</b> | 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.                           |
| <b>QUAL</b>                     | This command is entered only from the breakpoints panel. It displays the breakpoint qualification data to the user.  |
| <b>RESUME (RES)</b>             | 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 85).

- 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.
- To invoke a program, use the following fields, under the heading **INVOKE PROGRAM**:
  - PGM** The name of a program to be invoked as a dialog function.
  - PARM** Optional parameters to be passed to the program.



You can also specify:

**NEWAPPL** Indication of whether a new application is being invoked.

**yes, y** This function is a new application.

**no, n** This function is not a new application.

**ID** A 1- to 4-character ID for a new application. If you invoke a new application and leave the ID field blank, the default ID of ISP is used. Note that the ID determines the names of the profile and the command table to be used for the application.

---

```
----- INVOKE DIALOG FUNCTION/SELECTION MENU -----  
COMMAND ==>  
  
INVOKE SELECTION MENU:  
  PANEL ==> _                               OPT ==>  
  
INVOKE COMMAND:  
  CMD ==>  
  
INVOKE PROGRAM:  
  PGM ==>                                     PARM ==>  
  
NEWAPPL ==> NO                               ID ==>  
  
NEWPOOL ==> NO
```

Figure 85. Dialog Test - Invoke Function Entry Panel

---

**NEWPOOL** Indication of whether a new shared variable pool is to be created. This value is ignored if the value for NEWAPPL is "yes" or "y."

**yes, y** Create a new shared variable pool.

**no, n** Do not create a new shared variable pool.

Be aware of the following items when using the functions option:

**OPT and PARM** The OPT and PARM parameter input fields each continue on the following line.

**REQUIRED FIELDS** To invoke a function, you must specify a value for either the PANEL, CMD, or PGM parameter; but you may not specify more than one of them.

## PANELS (OPTION 7.2)

The panels option allows you, during panel development, to test newly created or modified panels and messages without having to build supporting code to display them. Any variables referenced and set during panel processing are handled according to standard ISPF protocol. When you select the panels option, you are shown a panel (Figure 86).

---

```
----- DISPLAY PANEL -----  
COMMAND ==>  
  
PANEL NAME ==>  
  
MESSAGE ID ==> (Optional)  
  
CURSOR ==> (Optional)
```

Figure 86. 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).

These are the same parameters that may be specified (from a dialog function) when invoking the DISPLAY service.

If you specify a message id without identifying a panel name, the message is displayed on a separate panel (Figure 87). 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 or until you leave the display panel option.

---

```
----- MESSAGE DISPLAY PANEL ----- SHORT MESSAGE
COMMAND ==> _
```

This panel is used to display your specified message.

Figure 87. 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 88) indicates all the current variables for the dialog being tested.

---

----- VARIABLES -----LINE 000001 COL 001 080  
COMMAND ==> SCROLL ==> PAGE

ADD AND CHANGE VARIABLES. UNDERSCORES NEED NOT BE BLANKED.  
ENTER END COMMAND TO FINALIZE CHANGES.

| VARIABLE      | P | A | VALUE  |
|---------------|---|---|--|
| '''' A1_____  | F |   | FUNCTION POOL VALUE FOR A1                                 |
| '''' A2_____  | F |   | FUNCTION POOL VALUE FOR A2                                 |
| '''' A5_____  | F |   | FUNCTION POOL VALUE FOR A5                                 |
| '''' A7_____  | F |   | FUNCTION POOL VALUE FOR A7                                 |
| '''' A8_____  | F |   | FUNCTION POOL VALUE FOR A8                                 |
| '''' A1_____  | S |   | SHARED POOL VALUE FOR A1                                   |
| '''' A2_____  | S | T | SHARED POOL VALUE FOR A2 WHERE THE VALUE HAS BEEN TRUNCATE |
| '''' A4_____  | S |   | SHARED POOL VALUE FOR A4                                   |
| '''' A10_____ | S |   | SHARED POOL VALUE FOR A10                                  |
| '''' B2_____  | S |   | SHARED POOL VALUE FOR B2                                   |
| '''' B3_____  | S | N | NON-MODIFIABLE SHARED POOL VALUE FOR B3                    |
| '''' B4_____  | S |   | SHARED POOL VALUE FOR B4                                   |
| '''' A5_____  | P |   | PROFILE POOL VALUE FOR A5                                  |
| '''' A9_____  | P | N | NON-MODIFIABLE PROFILE POOL VALUE FOR A9                   |
| '''' B1_____  | P |   | PROFILE POOL VALUE FOR B1                                  |
| '''' B2_____  | P |   | PROFILE POOL VALUE FOR B2                                  |

Figure 88. Dialog Test - Variables Entry Panel

---

Each line of the display represents a variable and contains:

'''' Line command area.

**VARIABLE** Name of the variable: an alphameric 8-character name, with the first character alphabetic. The alphabetic characters are translated to uppercase when you press the ENTER key. This field is required.

**P** Pool in which the variable exists, a required one-character field:

**F** Function variable pool

**S** Shared variable pool

**P** Profile variable pool

- A** Attributes of the variable, a nonmodifiable one-character field:
- N** Nonmodifiable variable. Some system-reserved variables are not modifiable.
- T** Truncated variable value. The actual value is longer than the 58 characters displayable on the panel.

**Note:** If you change a line showing a truncated value, only the data shown on the display is stored as the new value.

**VALUE** Value of the variable up to 58 characters.

The variables option uses the CANCEL, END, and LOCATE dialog test primary commands, and the I dialog test line command. These commands are described in the section entitled "Dialog Test Commands."

Normally, the variable pools are updated with the data from the display when you use the END command to leave the option.

## Manipulating Variables

The rows of the display are ordered by the pool containing the variables, then alphabetically by variable name within each pool. The function variable pool is listed first, followed by the shared variable pool, and then the profile variable pool. Insertions are left where they are entered on the display.

Modifications to the display are processed when you press the ENTER key. Updating of the variable pools occurs when you enter the END command.

You can create new dialog variables, but you cannot create two variables with the same name in the same variable pool. Also, you cannot delete a variable, but you can set its value to nulls.

To create a new variable, you may either:

- Use the I (insert) line command to create a new row and then enter the variable name, pool, and value on the new line. For each field, move the cursor to the start of the field and enter new information. The underscores are pad characters and need not be blanked out.
- Take an existing variable and change its name, its pool, or both. This creates a new variable with the old variable's last value and resets the old variable's value to nulls.

By using the second method, you can interchange the values of two or more variables by simply changing their names. For example, you can interchange the values for variables A and B by changing the variable name A to B and name B to A and then pressing ENTER.

## Variables Usage Notes

When using the variables option, you should be aware of the following:

- TEST MODE** Variable manipulations performed under dialog test at a breakpoint are considered an extension of your dialog and so are handled in user mode. Dialog variables, table data, and service return codes that you introduce, delete, or change are treated as though your dialog had made those changes.
- VARIABLE LIFE** Profile variables that you create remain in your profile pool from one dialog test session to another. Shared and function variables exist only for the duration of dialog test.
- VARIABLE SEQUENCE** Any changes you make on the scrollable display are left where you enter them. Sorting is performed only when you reselect the variables option.
- SPLIT-SCREEN MODE** In split screen mode, a profile variable pool may be shared between two logical screens. Since the variables option takes a snapshot of the variables, any change to a profile variable on one screen is not immediately reflected on the other screen. To get the latest changes, reselect the variables option. Also, when one profile variable is changed on two logical screens using split screen mode, the changed profile variable on the screen where the last END command was entered takes precedence.
- VARIABLE VALUE** Variables defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command. When a VDEFINE error occurs, a panel identifies the data and its value and describes the error. You must then correct the error and press the ENTER key. If you create a new variable by changing the pool indicator of an existing variable defined as other than character, the new variable will have character format.
- HEXADECIMAL DATA** Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.

## TABLES (OPTION 7.4)

The tables option allows you to examine and manipulate the rows of a table, and to display table structure and status. When you select this option, a panel is displayed (Figure 89) on which you indicate the table function you want and the parameters needed to identify the table.

---

```
----- TABLES -----
OPTION ==> _

1  Display row           5  Display structure
2  Delete row           6  Display status
3  Modify row
4  Add after row

TABLE NAME      ==>          CURRENT ROW:

ROW IDENTIFICATION:
  BY ROW NUMBER ==> *      (*=Current row)

  BY VARIABLE    VALUE      (Search for row if row number blank)

  _____
  _____
  _____
  _____
  _____
  _____
```

Figure 89. Dialog Test - Tables Selection Panel

---

Use the fields on the tables entry panel as follows:

|               |   |
|---------------|---|
| <b>OPTION</b> | Enter the number of the table function: |
| 1             | Display row                             |
| 2             | Delete row                              |
| 3             | Modify row                              |
| 4             | Add after row                           |
| 5             | Display structure                       |
| 6             | Display status                          |



|                           |   |
|---------------------------|---|
| <b>TABLE NAME</b>         | 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.   |
| <b>CURRENT ROW</b>        | Displays the position number of the current row after you have identified a table. This field is nonmodifiable.   |
| <b>ROW IDENTIFICATION</b> | Identify a row, either directly by row number or indirectly by specifying a search argument of table variable names and their values.   |
| <b>BY ROW NUMBER</b>      | Enter the position number of the table row desired or * for the current row. If you are adding a row, you can use: <ul style="list-style-type: none"> <li><b>top, t, 0</b>                    Make the new row first in the table.</li> <li><b>bottom, bot, b</b>            Make the new row last in the table.</li> </ul> |
| <b>BY VARIABLE</b>        | 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.  |
| <b>VALUE</b>              | 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 (*).   |

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 90). In the figure, the variables constitute one table row.

---

```
DISPLAY ROW TABLE SAMTB1 ROW 3 -----LINE 000001 COL 001 080
COMMAND ==> SCROLL ==> PAGE

VARIABLE T A VALUE
K1_____ K This is the value for key K1
K2_____ K My value for K2
K3_____ K This is the value to be used
N1_____ N The first name value is this field
N2_____ N Name 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 90. 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 91) 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 91. Dialog Test - Delete Table Row Confirmation 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.
- You may not change the variable name for a key variable or name variable; if you do, an error message is displayed and the original name is restored.
- You may not delete a key or name variable and its value from the display or table row.
- If you delete a save variable, it is assumed that the variable no longer exists in this row.
- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable (closest to the bottom of the scrollable display).
- If the table has keys, the values for the keys in the added row must be unique among the existing rows when you leave the modify row option. Otherwise, a message is displayed and the row is redisplayed so that you can change the keys.
- Blank save names are ignored and need not be deleted, even if data is left in the value.
- Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.
- Variables that have been defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command.

- When you leave the modified row option using the END command, the row is replaced and the message "ROW MODIFIED" is issued.

The modify row option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. These commands are described in the section entitled "Dialog Test Commands." Inserted and repeated lines always have a type of S since only save variables may be added to a row of an existing table. Only save variables may be deleted.

## Add After Row (Tables Option 4)

The add after row option is used to add a new row after a selected row of an opened table. It is selected when you enter 4 on the option line of the tables panel after naming an open table and identifying the row of the table that the new row will follow.

When you select this option, a scrollable display is shown (Figure 93), 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.





- Blank save names are ignored and need not be deleted, even if data is left in the value.
- Nondisplayable hexadecimal data is displayed or entered using the form X'nnnnnnnn', where n is an integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotes.
- Variables that have been defined (using the VDEFINE service) as other than character are shown in translated form; do not modify them using the hexadecimal representation. A format or length error causes an error message to be displayed when you use the END command.

The add after row option uses the CANCEL, END, and LOCATE dialog test primary commands, and the D, I, and R dialog test line commands. Refer to "Dialog Test Commands." Inserted and repeated lines always have a type of S since only save variables may be added to a row of an existing table. Only save variables may be deleted.

## Display Structure (Tables Option 5)

When you select option 5 from the tables option panel, you are shown a scrollable display (Figure 94) with the table name (in this case, SAMTB1) in the panel header.

The display shows the following fields:

|                            |   |
|----------------------------|---|
| <b>NUMBER OF KEYS</b>      | Number of key variables in a row              |
| <b>NUMBER OF NAMES</b>     | Number of name variables in a row             |
| <b>NUMBER OF ROWS</b>      | Number of rows currently in the table         |
| <b>CURRENT ROW POINTER</b> | Current row pointer value                     |
| <b>KEYS</b>                | A list of the names of all the key variables  |
| <b>NAMES</b>               | 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 94. 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 95).

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

**TIME CREATED** Time the table was created

---

----- STATUS FOR TABLE SAMTB1 -----

|                        |            |                     |            |
|------------------------|------------|---------------------|------------|
| STATUS FOR THIS SCREEN | : NOT OPEN | DATE CREATED        | : 81/06/15 |
| TABLE AVAILABLE        | : YES      | TIME CREATED        | : 10:15    |
|                        |            | LAST DATE MODIFIED: | 81/06/25   |
|                        |            | LAST TIME MODIFIED: | 14:52      |
|                        |            | LAST MODIFIED BY    | : Z59SBL   |
|                        |            | ORIGINAL ROW COUNT: | 50         |
|                        |            | CURRENT ROW COUNT   | : 60       |
|                        |            | MODIFIED ROW COUNT: | 15         |
|                        |            | UPDATE COUNT        | : 4        |

Figure 95. Dialog Test - Table Status Display for a Closed Table

---

|                           |  |
|---------------------------|--|
| <b>LAST DATE MODIFIED</b> | Date the table was last modified   |
| <b>LAST TIME MODIFIED</b> | Time the table was last modified   |
| <b>LAST MODIFIED BY</b>   | User id of the user who last modified the table  |
| <b>ORIGINAL ROW COUNT</b> | The number of rows that were added to a newly created table before closing the table for the first time. |

|                           |  |
|---------------------------|--|
| <b>CURRENT ROW COUNT</b>  | The number of rows currently in the table.   |
| <b>MODIFIED ROW COUNT</b> | 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. |
| <b>UPDATE COUNT</b>       | Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.                    |

**Note:** Tables option 4.3 (modify row) allows you to modify a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when modifying a key value.

If the table is open for your user id, you are shown a status panel (Figure 96). 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

**TIME CREATED**

Time the table was created

**LAST DATE MODIFIED**

Date the table was last modified

**LAST TIME MODIFIED**

Time the table was last modified

**LAST MODIFIED BY**

User id of the user who last modified the table

**ORIGINAL ROW COUNT**

The number of rows that were added to a newly created table before closing the table for the first time.

**CURRENT ROW COUNT**

The number of rows currently in the table.

## MODIFIED ROW COUNT

The number of rows in the table that have been modified at least once. A row that has been added to an existing table is also considered a modified row.

## UPDATE COUNT

Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

**Note:** Tables option 4.3 (modify row) allows you to modify a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when modifying a key value.

---

----- STATUS FOR TABLE SAMTB1 -----

|                           |          |                     |            |
|---------------------------|----------|---------------------|------------|
| STATUS FOR THIS SCREEN    | : OPEN   | DATE CREATED        | : 81/06/15 |
| OPEN OPTION               | : WRITE  | TIME CREATED        | : 10:15    |
| TABLE ON DISK             | : YES    | LAST DATE MODIFIED: | 81/06/25   |
| LAST TABLE SERVICE        | : TBSCAN | LAST TIME MODIFIED: | 14:52      |
| LAST SERVICE RETURN CODE: | 0        | LAST MODIFIED BY    | : Z59SBL   |
| CURRENT ROW POINTER       | : 15     | ORIGINAL ROW COUNT: | 50         |
|                           |          | CURRENT ROW COUNT:  | 60         |
|                           |          | MODIFIED ROW COUNT: | 15         |
|                           |          | UPDATE COUNT        | : 4        |

Figure 96. 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 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 (ISPLNK) calls are displayed with their parameter values separated by commas. Name lists are shown as coded in the dialog, in string format or structure format. (Structure format includes the count, element length, and list of names.) For variable services parameters whose context is defined by the "name-list" parameter on the service call, the first four bytes of the parameter value are displayed in hexadecimal (X'nnnnnnnn').
- Dialog test calls are shown using the command call format without the ISPEXEC prefix.

## Variable Trace Entries

Two variable trace entry lines are generated for each variable trace log entry. The variable may be referenced or set by a user dialog (directly or indirectly through the use of a dialog service), or explicitly set by a dialog test option on behalf of a user. The summary parts of these entry lines identify the trace. Line one shows the name of the variable, the pool in which it exists (F for function, S for shared, P for profile), and an indicator (TEST) if a dialog test option set the value. Line 2 shows the operation performed for the variable (GET, PUT, or CHG) and the name of the dialog service that performed the operation (for non-TEST entries). The current value of the variable is printed in the detail section of the log entry and can span two lines; for example:

```
LIB1.... POOL(P) .... - VALUE(FLAG)
...GET by EDIT      -
```

**Note:** The value is truncated after the second line.

If the variable value contains nondisplayable characters, the value is displayed in hexadecimal format (X'nnnnnnnn').

## 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 97 shows the invoke dialog service panel.

---

```
----- INVOKE DIALOG SERVICE -----
COMMAND ==>

ENTER DIALOG SERVICE AND ITS PARAMETERS:
==> _
```

Figure 97. 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 98).

---

```
----- SPECIAL DISPLAY PANEL -----  
COMMAND ==> _
```

This panel is used for two special DISPLAY conditions:

1. When DISPLAY is invoked with a message, but without a panel name.
2. When DISPLAY is invoked without any parameters.

Figure 98. 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 99) 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 99. Dialog Test - Traces Selection Panel

---

### Function Trace (Trace Option 1)

The function trace option is used to establish criteria for recording the names of dialog service calls, the service parameters, and return code in the ISPF log. Service calls made by the dialog or during test processing (e.g., using dialog test option 2 to display a panel) are recorded. Whenever a new application and/or function has data recorded, a header is placed in the trace.

When you select the function trace option, you are shown a scrollable panel (Figure 100) that lists all currently defined function traces.



## 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 101) 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.



|                  |   |
|------------------|---|
| <b>OPERATION</b> | Type of variable reference to trace:  |
| <b>get, g</b>    | Accesses to the variable's value  |
| <b>put, p</b>    | Stores to the variable's value  |
| <b>chg, c</b>    | Changes to the variable's value   |
| no entry         | All references to variable are traced   |
| <b>FUNCTION</b>  | The user dialog function for which the trace is defined. If no entry, indicates this variable will be traced for all functions. |
| <b>ACTIVE</b>    | Indication of whether the trace is to be active:  |
| <b>yes, y</b>    | The trace is currently active   |
| <b>no, n</b>     | 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 102), 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  |
| <b>SERVICE</b>  | Name of the dialog service at which to interrupt dialog execution. This field is required.   |
| <b>WHEN</b>     | Indication of when the breakpoint should occur:<br><br><b>before, b</b> Before the service receives control<br><br><b>after, a</b> After the service completes execution<br><br><b>rnn</b> After the service completes execution, if the return code is the integer nn<br><br><b>no entry</b> Before and after service execution |
| <b>FUNCTION</b> | 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.   |
| <b>ACTIVE</b>   | Indication of whether the breakpoint is to be active at this time:<br><br><b>yes, y</b> It is currently active<br><br><b>no, n</b> It is currently not active<br><br><b>no entry</b> It is currently active  |
| <b>*QUAL*</b>   | If present at the end of a row, indicates that qualification data exists for the breakpoint. This field is nonmodifiable. For additional information see "Qualifying a Breakpoint," below.   |

Empty lines (all input fields contain underscores) are added to the initial display to fill up the screen. If you delete all the lines used for defining breakpoints, the display is automatically refreshed with enough empty lines to fill the screen again.







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:

- |                        |  |
|------------------------|--|
| <b>QUALIFICATION</b>   | If you plan to qualify several breakpoints, it may be more efficient to specify all breakpoint data on the qualification panel.  |
| <b>END COMMAND</b>     | You can use the END primary command from either the breakpoints panel or the breakpoints qualification panel.  |
| <b>INPUT ERRORS</b>    | 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. |
| <b>SYNTAX CHECKING</b> | A dialog service call must pass a basic syntax check before a breakpoint will be honored.  |
| <b>CONTROL DISPLAY</b> | 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 104 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 104. Dialog Test - Breakpoint Primary Option Menu

---

Like the dialog test primary option menu, the breakpoint primary option menu allows you to use the RETURN command from any one of the selected test options to process a redisplay of the breakpoint primary option menu. At the breakpoint primary option menu, the END and RETURN commands have no effect. You must use the GO option to terminate processing at this breakpoint and continue executing the dialog being tested, or the CANCEL option to cancel the dialog test option. This protects against inadvertent loss of data.

The breakpoint primary option menu contains all of the options of the dialog test primary option menu, and as such, presents all of the dialog test functions to you. It also displays the following information:

**Panel Title** BREAKPOINT PRIMARY OPTION MENU

**BEFORE or AFTER** An indication of whether the dialog has been suspended BEFORE or AFTER the service has executed.

**Service Name** The name of the service at which the dialog has been suspended. In Figure 104, the service name is **DISPLAY**.

**CURRENT STATUS:**

**APPLICATION** The application identifier of the suspended user dialog.

**FUNCTION** The program or command name of the suspended user dialog.

**BREAKPOINT** Up to two display lines showing an image of the dialog service call. Truncation occurs after the second line.

ISPEXEC calls are shown as coded.

ISPLINK (ISPLNK) calls are displayed with their parameter values separated by commas. Name lists are shown as coded in the dialog, in string format or in structure format (structure format includes the count, element length, and list of names). For variable services parameters whose context is defined by the "name-list" parameter on the service call (for example, the variable value areas for a VDEFINE), the first four bytes of the parameter value are displayed in hexadecimal (X'nnnnnnnn').

**RETURN CODE** The dialog service return code. This field is displayed only if the breakpoint occurs **AFTER** the dialog service has executed. The return code field is modifiable; its value is passed back to the dialog (as the service's) when you select the GO option. This facilitates the testing of dialog error handling.

**GO** The GO option continues dialog execution from a breakpoint. The user dialog resumes processing from the point at which it was suspended.

**CANCEL** The CANCEL option terminates dialog testing and redisplay the first primary option menu you

displayed at the beginning of your ISPF session. All trace and breakpoint definitions are lost when dialog test is terminated.

When a user dialog encounters a breakpoint, the current dialog environment is saved. When you select the GO option, the environment is restored, except for the following:

- If you change variable, table, and file tailoring data at a breakpoint, these actions are performed as an extension of the suspended dialog; it is as though the dialog had taken all the actions itself during execution.
- If you modify the service return code (on the breakpoint primary option menu), the new return code is passed back to the dialog (as though the service had set the new code itself).
- If you execute the PANELID command at the breakpoint, the last setting for displaying panel identifiers is retained.
- If any CONTROL service settings for DISPLAY LINE or DISPLAY SM were in effect before the breakpoint, such settings are lost.

Note that the manipulation of one dialog part may cause a change to another dialog part. For example, if a panel is displayed, variables may be set.

All trace and breakpoint definitions are lost if you select the CANCEL option from the breakpoint primary option menu.

## CHAPTER 12. TUTORIAL (OPTION T)

The tutorial option provides immediate online reference and instruction on how to use PDF. You can invoke it by selecting option T from the primary option menu or by using the HELP command. You can scan the tutorial sequentially from beginning to end, or you can select specific topics from an alphabetic index or table of contents.

When you invoke the tutorial from the primary option menu, introductory pages are displayed to explain how the tutorial is organized and how it might be used (Figure 105). 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 105 (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 105 (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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID (YOUR TSO USERID)<br>TYPE : ASM                             | 5. ALLOCATE NEW DATA SET PANEL   |
| 6. ENTER THE FOLLOWING:<br>VOLUME : (BLANK)<br>UNITS : TRKS<br>PRIMARY : 2<br>SECONDARY: 1<br>DIR BLKS : 1<br>FORMAT : FB<br>LENGTH : 80<br>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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID (YOUR TSO USERID)<br>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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID<br>TYPE : ASM<br>MEMBER : (BLANK)  | 13. MEMBER ISRASM IS COPIED<br>FROM DATA SET SYS1.SAMPLIB<br>TO SAMPLE.USERID.ASM.<br>THE FIRST MOVE/COPY PANEL<br>IS REDISPLAYED. |
| 14. PRESS PF3   | 14. ISPF/PDF PRIMARY OPTION MENU   |
| 15. SELECT BROWSE (OPTION 1)  | 15. BROWSE ENTRY PANEL   |
| 16. SPECIFY ISPF LIBRARY:<br>PROJECT : SAMPLE<br>LIBRARY : USERID<br>TYPE : ASM<br>MEMBER : (LEAVE BLANK) | 16. BROWSE MEMBER LIST OF<br>DATA SET SAMPLE.USERID.ASM  |
| 17. SELECT MEMBER ISRASM<br>(ENTER "S" AHEAD OF MEMBER NAME)  | 17. BROWSE TEXT DISPLAY OF<br>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:<br>==> FIND COMMENT  | 20. STRING "COMMENT" IS<br>INTENSIFIED.  |
| 21. FIND NEXT OCCURRENCE OF "COMMENT"<br>BY PRESSING PF5  | 21. SECOND OCCURRENCE OF<br>"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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID<br>TYPE : ASM<br>MEMBER : ISRASM        | 26. EDIT DATA DISPLAY<br>OF DATA SET<br>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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID<br>TYPE : ASM<br>MEMBER : ISRASM<br>LIST ID : ASMTEST1<br>ASSEMBLER<br>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:<br>PROJECT : SAMPLE<br>LIBRARY : USERID<br>TYPE : ASM<br>MEMBER : ISRASM<br>LISTID : ASMTEST2<br>ASSEMBLER<br>PARAMETERS: OBJ,NODECK,LIST,TERM | 43. JCL FOR ASSEMBLING ISPASM 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:<br>==> 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. |

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.

|  |  |
|--|--|
| 49. (AT PRIMARY OPTION MENU)<br>PRESS PF3  | 49. TERMINATION PANEL  |
| 50. SELECT OPTION "J" 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. Source listings may also be obtained via the Print option of the PDF library maintenance utility and the move/copy utility. PDF provides index listings, upon user request, via the Index option of the library maintenance 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 106. *(page 237)*

Information at the top of the page includes project, library, 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 107 . 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 108 . Here, the module attributes are printed to the right of each member name.



Finally, a sample ISPF log listing is shown in Figure 109. 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.

PROJECT: SPFDEMO  
LIBRARY: MYLIB  
TYPE: PLI

MEMBER: COINS  
LEVEL: 01.04  
USERID: JOSLIN

DATE: 79/05/18  
TIME: 17:22  
PAGE: 01 OF 01

COINS

| START COL | 1   | 2                         | 3                   | 4 | 5 | 6 | 7 | 8        | MOD 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 106. Sample Source Listing

PROJECT: SPFDEMO  
 LIBRARY: MYLIB  
 TYPE: PLI

DATE: 82/01/06  
 TIME: 14:10  
 PAGE: 001

GENERAL DATA:  
 VOLUME SERIAL: D00163  
 DEVICE TYPE: 3330  
 ORGANIZATION: PO  
 CREATION DATE: 80/11/07

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 | VERS.MOD LEVEL | CREATION DATE | DATE AND TIME LAST MODIFIED | CURRENT NO. LINES | INITIAL NO. LINES | MODIFIED NO. LINES | USER ID |
|-------------|----------------|---------------|-----------------------------|-------------------|-------------------|--------------------|---------|
| ACCOUNT     | 01.00          | 81/01/09      | 79/01/09 17:07              | 21                | 21                | 0                  | HUNTER  |
| ACCT1       | 01.01          | 81/01/09      | 79/04/23 14:52              | 99                | 193               | 0                  | BECKETT |
| ACCT2       | 01.00          | 81/01/09      | 79/01/09 17:07              | 20                | 20                | 0                  | BECKETT |
| COINS       | 01.04          | 81/04/24      | 79/04/28 16:20              | 19                | 19                | 4                  | JOSLIN  |
| COMPX       | 01.00          | 81/01/09      | 79/01/09 17:08              | 44                | 44                | 0                  | JOSLIN  |
| COMPY       | 01.01          | 81/01/14      | 79/01/16 12:30              | 13                | 13                | 1                  | JOSLIN  |
| DCLS        | 01.00          | 81/04/23      | 79/04/23 15:14              | 20                | 20                | 0                  | LEESB   |
| LISTOUT     | 01.02          | 81/04/23      | 79/05/06 15:00              | 17                | 13                | 6                  | LEESB   |
| MAIN        | 01.00          | 81/01/09      | 81/01/09 17:08              | 4                 | 4                 | 0                  | KIRK    |
| TESTDIR     | 01.02          | 81/04/23      | 81/05/06 17:04              | 30                | 43                | 10                 | LAIDLAW |
| MAXIMUMS:   | 01.04          | 81/04/24      | 81/05/06 17:04              | 99                | 193               | 10                 |         |
| TOTALS:     |                |               |                             | 287               | 390               | 21                 |         |

END OF MEMBER LIST

Figure 107. Sample Index Listing - Source Library

PROJECT: SPFDEMO  
 LIBRARY: MYLIB  
 TYPE: LOAD

DATE: 82/01/08  
 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: 81/11/05 | 1ST EXTENT SIZE: 44 |                     | 8 MEMBERS            |
|                         | SECONDARY QUAN: 4   |                     |                      |

| MEMBER NAME | LOAD (HEX) | MODULE SIZE (DEC) | ENTRY POINT (HEX) | TTR (HEX) | ALIAS NAME | AUTH CODE | MODULE ATTRIBUTES     |
|-------------|------------|-------------------|-------------------|-----------|------------|-----------|-----------------------|
| SPFEDIT     | 000148     | 328               | 000000            | 000006    |            | 00        | F-ONLY REFR RENT REUS |
| SPFMAIN     | 004F98     | 20,376            | 000000            | 00000D    |            | 00        | F-ONLY REFR RENT REUS |
| SPFPMD      | 0016E8     | 5,864             | 000000            | 000204    |            | 00        | F-ONLY REFR RENT REUS |
| SPFSUBS     | 024698     | 149,144           | 000000            | 000211    |            | 00        | F-ONLY REFR RENT REUS |
| SPFTBLS     | 002470     | 9,328             | 000000            | 00110B    |            | 00        | F-ONLY                |
| SPFUDA      | 0046D8     | 18,136            | 000000            | 00120A    |            | 00        | F-ONLY REFR RENT REUS |
| SPFURS      | 000FC0     | 4,032             | 000000            | 00140A    |            | 00        | F-ONLY REFR RENT REUS |
| SPF3277     | 000548     | 1,352             | 000000            | 001504    |            | 00        | F-ONLY REFR RENT REUS |
| TOTALS:     |            | 00032EB0          | 208,560           |           |            |           |                       |

END OF MEMBER LIST

-----  
 MODULE ATTRIBUTE CODES:  
 F-ONLY PROCESS ONLY BY F LEVEL LINKAGE EDITOR  
 NON-EXEC NOT EXECUTABLE  
 OL ONLY LOADABLE  
 OVLY IN OVERLAY STRUCTURE  
 REFR REFRESHABLE  
 RENT REENTERABLE  
 REUS REUSABLE  
 SCTR SCATTER FORMAT  
 TEST MODULE TO BE TESTED

Figure 108. Sample Index Listing - Load Library

```
TIME          *** SPF TRANSACTION LOG ***                USERID: FRANK3    DATE: 81/12/22    PAGE: 1
17:11  START OF ISPF SESSION LOG #64 -----
17:12  UTILITY - ALLOCATE - SPFDemo.NEWLIB.PLI ALLOCATED ON VOLUME WORK97
17:13  UTILITY - MOVE/COPY- SPFDemo.MYLIB.PLI(Account) COPIED
17:13  TO SPFDemo.NEWLIB.PLI(ACCTXX)
17:13  UTILITY - MOVE/COPY- SPFDemo.MYLIB.PLI(ACCT1) COPIED
17:13  TO SPFDemo.NEWLIB.PLI(ACCTYY)
17:13  UTILITY - MOVE/COPY- SPFDemo.MYLIB.PLI(ACCT2) COPIED
17:13  TO SPFDemo.NEWLIB.PLI(ACCTZZ)
17:13  UTILITY - MOVE/COPY- SPFDemo.MYLIB.PLI(COINS) COPIED
17:13  TO SPFDemo.NEWLIB.PLI(COINS)
17:14  EDIT    - SAVE    - SPFDemo.NEWLIB.PLI(COINS) - MEMBER SAVED
17:15  EDIT    - CREATE  - SPFDemo.NEWLIB.PLI(ACCTNEW) - MEMBER CREATED
17:15  EDIT    - SAVE    - SPFDemo.NEWLIB.PLI(ACCTXX) - MEMBER SAVED
17:15  EDIT    - SAVE    - SPFDemo.NEWLIB.PLI(ACCTYY) - MEMBER SAVED
17:16  EDIT    - SAVE    - SPFDemo.NEWLIB.PLI(ACCTZZ) - MEMBER SAVED
17:17  UTILITY - DELETE  - SPFDemo.NEWLIB.PLI(ACCTZZ) DELETED
17:17  TSO     - COMMAND - LISTC
17:17  UTILITY - ALLOCATE - SPFDemo.NEWLIB.OBJ ALLOCATED ON VOLUME WORK98
17:19  JOB    - FLAG8   - PLI OPT COMPILE -- TO SPFDemo.NEWLIB.OBJ(COINS)
17:19  JOB    - FLAG8   - SUBMITTED
17:20  END OF ISPF SESSION LOG #64 -----
```

Figure 109. Sample Log 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 110 and Figure 111 .

|    | 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        | ↑ | ↓  | ≤ | [ | L | → | 8F |
| 90 | □  | j        | k        | l        | m        | n        | o        | p        | q        | r        | ∩ | ∪  |   | o |   | ← | 9F |
| A0 | -  | ~        | s        | t        | u        | v        | w        | x        | y        | z        | ∩ | ∪  | ⊥ | [ | ≥ | ° | AF |
| B0 | α  | ε        | ι        | ρ        | ω        |          | ×        | \        | ÷        |          | ∇ | Δ  | τ | ] | ≠ |   | BF |
| C0 | {  | A        | B        | C        | D        | E        | F        | G        | H        | I        | ∞ | ∞  |   | φ |   | ∞ | CF |
| D0 | }  | J        | K        | L        | M        | N        | O        | P        | Q        | R        | I | !  | ∇ | Δ | ∩ | ∪ | DF |
| E0 | \  |          | S        | T        | U        | V        | W        | X        | Y        | Z        | + | +  |   | e | ⊗ | ⊗ | EF |
| F0 | 0  | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        |   | ∇  | Δ | ⊗ | ⊗ |   | FF |

 3278 only; invalid character on 3277.


 National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Figure 110. Internal Character Representations for APL Keyboards

|    | 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B  | C | D | E | F |    |
|----|----|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|----|
| 00 |    |   |   |   |   |   |   |   |   |   |   |    |   |   |   |   | 0F |
| 10 |    |   |   |   |   |   |   |   |   |   |   |    |   |   |   |   | 1F |
| 20 |    |   |   |   |   |   |   |   |   |   |   |    |   |   |   |   | 2F |
| 30 |    |   |   |   |   |   |   |   |   |   |   |    |   |   |   |   | 3F |
| 40 | sp |   |   |   |   |   |   |   |   | ¢ | . | <  | ( | + |   |   | 4F |
| 50 | &  | 1 | 2 | 3 |   |   |   |   |   | ↓ | ! | \$ | * | ) | ; | - | 5F |
| 60 | -  | / |   |   |   |   |   |   |   |   | , | %  | _ | > | ? |   | 6F |
| 70 | n  | ° |   |   |   |   |   |   |   | ' | : | #  | @ | ' | = | " | 7F |
| 80 |    | a | b | c | d | e | f | g | h | i | ↑ | {  | ≤ | ( | + | + | 8F |
| 90 | □  | j | k | l | m | n | o | p | q | r |   | }  | ≠ | ) | ± | ■ | 9F |
| A0 | -  | ~ | s | t | u | v | w | x | y | z | • | L  | Γ | [ | ≥ | • | AF |
| B0 | o  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ∇ | ┘  | └ | ] | ≠ | - | BF |
| C0 | {  | A | B | C | D | E | F | G | H | I | Δ | ⊥  | T | V | + | + | CF |
| D0 | }  | J | K | L | M | N | O | P | Q | R | ⊞ | Δ  | § | ¶ | ← | → | DF |
| E0 | \  | \ | S | T | U | V | W | X | Y | Z | ∇ | †  | ‡ | ♣ | ♠ | ^ | 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 111. 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.

## MEMBER LIST COMMANDS

The following commands may be entered in the command input field (line 2 of the logical screen) on member list displays.

**LOCATE** Causes direct scrolling to the specified member name (may be entered on any member list display).

**SELECT** Causes the specified member name to be selected for browse, edit, foreground, or batch (invalid on utility member list displays). Allows selection of a member which is not in the list.

On member list displays, the following one-character commands may be entered at the beginning of a line, ahead of the member name.

**B (browse)** Causes the member to be browsed (valid only under the library utility).

- D (delete)** Causes the member to be deleted (valid only under the library utility).
- P (print)** Causes the member to be printed (valid only under the library utility).
- R (rename)** Causes the member to be renamed (valid only under the library utility). The new name must be entered to the right of the member name.
- S (select)** Causes the member to be selected for browse, edit, foreground, batch, and for move/copy, reset, or convert utility (invalid under the library utility).

## BROWSE COMMANDS

The following commands may be entered in the command input field (line 2 of the logical screen) under browse.

- CAPS** Causes character strings, entered as part of the FIND command, to be converted to uppercase. CAPS OFF causes character strings to be left as-is.
- COLS** Causes display of a line that identifies column positions.
- FIND** Causes the data to be scanned for one or all occurrences of a character string. The cursor is repositioned to the beginning of the first occurrence, if it is found.
- HEX** Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.
- LOCATE** Causes direct scrolling to the specified relative line number or label.
- RESET** Removes the column identification line from display.
- .xxxxx** Establishes a label (PDF internal symbol) which is equated to the top line on the screen. It can then be used with the LOCATE command to cause scrolling directly to that line.

## EDIT PRIMARY COMMANDS

Under edit, the following commands may be entered in the command 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.

**CANCEL** Causes editing to be terminated without saving the data.

**CAPS** Causes alphabetic data entered from the terminal to be translated to uppercase, including FIND and CHANGE strings. CAPS OFF causes alphabetic data to be left as-is.

**CHANGE** Causes one or all occurrences of a character string to be replaced with a second string. The cursor is repositioned to the end of the first occurrence, if it is found.

**COPY** Causes data to be copied from another source and inserted at a point indicated with the A (after) or B (before) line command. The other source can be a member of 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.

**FIND** Causes the data to be scanned for one or all occurrences of a character string. The cursor is repositioned to the beginning of the first occurrence, if it is found.

**HEX** Causes data to be displayed in hexadecimal format. HEX OFF restores normal display format.

**LOCATE** Causes direct scrolling to the specified line number.

**MODEL** Causes model data to be copied from a skeleton library and inserted at a point indicated with the A (after) or B (before) line command.

**MOVE** Same as COPY, except that the other source is deleted after the copy operation.

**NULLS** Replaces trailing blanks on the screen with null characters. NULLS OFF causes trailing blanks to be sent to the screen as blanks.

**NUMBER** Causes sequence numbers to be generated for any new lines that are created via insert, repeat, or copy. NUMBER OFF causes lines to be generated without numbers.

**PROFILE** Causes the display of the current setting of edit modes (NUMBER, AUTONUM, STATS, AUTOLIST, CAPS, NULLS, TABS, HEX, and RECOVERY). Also displays the MASK, TABS, and BOUNDS lines whenever they have settings other than their defaults.

|                 |  |
|-----------------|--|
| <b>RECOVERY</b> | 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. |
| <b>RENUM</b>    | Renumbers each line and turns on number mode.  |
| <b>REPLACE</b>  | Same as CREATE, except that it allows an existing member or an entire sequential data set to be replaced.  |
| <b>RESET</b>    | Causes a general resetting of intensified messages, incomplete line commands, and special lines.   |
| <b>SAVE</b>     | Causes the data to be stored back into the edit data set and editing to continue.  |
| <b>STATS</b>    | Causes statistics to be generated or updated whenever a member is saved, created, or replaced. STATS OFF causes members to be stored without statistics.   |
| <b>SUBMIT</b>   | Causes the data being edited to be submitted to the job stream for batch execution.  |
| <b>TABS</b>     | 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.                              |
| <b>UNNUM</b>    | Causes sequence numbers to be set to blanks and turns off number mode.   |

## EDIT LINE COMMANDS

Under edit, line commands may be entered at the beginning of a line by overtyping the line number.

|                   |  |
|-------------------|--|
| <b>A (after)</b>  | Identifies the line after which copied, moved, or model lines are to be inserted.        |
| <b>B (before)</b> | Identifies the line before which copied, moved, or model lines are to be inserted.       |
| <b>BOUNDS</b>     | Causes display of the column boundary line, to allow redefinition of boundary positions. |
| <b>COLS</b>       | Causes display of a line that identifies column positions.                               |
| <b>C (copy)</b>   | Identifies a line to be copied.  |
| <b>D (delete)</b> | Causes the line to be deleted.   |
| <b>F (first)</b>  | Causes the first line in a block of excluded lines to be redisplayed.                    |

|                     |  |
|---------------------|--|
| <b>I (insert)</b>   | Causes a new line to be inserted after this line.  |
| <b>L (last)</b>     | Causes the last line in a block of excluded lines to be redisplayed.   |
| <b>MASK</b>         | Causes display of the insert mask, to allow pre-entry of data on inserted lines.   |
| <b>M (move)</b>     | Identifies a line to be moved.   |
| <b>O (overlay)</b>  | 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). |
| <b>R (repeat)</b>   | Causes the line to be repeated and the duplicate copy to be inserted after this line.  |
| <b>S (show)</b>     | Causes an excluded line to be redisplayed.   |
| <b>TABS</b>         | Causes display of the tab line, to allow definition of hardware, software, and logical tab positions.  |
| <b>TE</b>           | Facilitates bulk text entry by allowing the user to type continuously without regard to the line number area.  |
| <b>TF</b>           | Causes reflowing of a paragraph of text from the cursor position to the end of the paragraph.  |
| <b>TS</b>           | Causes splitting of a line into two lines at the designated cursor position to facilitate the insertion of new material in existing text.  |
| <b>X (exclude)</b>  | Causes the line to be excluded from display. May also be used to limit the scope of the FIND and CHANGE commands.  |
| <b>&lt; (left)</b>  | Causes the contents of the line to be "data" shifted left.   |
| <b>&gt; (right)</b> | Causes the contents of the line to be "data" shifted right.  |
| <b>( (left)</b>     | Causes the contents of the line to be "column" shifted left.   |
| <b>) (right)</b>    | Causes the contents of the line to be "column" shifted right.  |

The following line commands may be entered with double characters to indicate the beginning and end of a block of lines.

|    |                 |    |                              |
|----|-----------------|----|------------------------------|
| CC | - Block copy    | << | - Block "data" shift left    |
| DD | - Block delete  | >> | - Block "data" shift right   |
| MM | - Block move    | (( | - Block "column" shift left  |
| OO | - Block overlay | )) | - Block "column" shift right |
| RR | - Block repeat  |    |                              |
| XX | - Block exclude |    |                              |

The following line commands may incorporate a number (n) consisting of one or more digits.

An - Insert n copies of the moved/copied line(s) after this line

Bn - Insert n copies of the moved/copied line(s) before this line

Cn - Copy n lines starting at this line

Dn - Delete n lines starting at this line

Fn - Show first n lines in a block of excluded lines

In - Insert n lines following this line

Ln - Show last n lines in a block of excluded lines

Mn - Move n lines starting at this line

On - Overlay n lines

Rn - Repeat this line n times

RRn - Repeat block of lines n times

Sn - Show n lines in a block of excluded lines

TEn - Insert n text entry lines following this line

TFn - Flow text from left bound to column position n

TSn - Insert n new lines between the split lines

Xn - Exclude n lines starting at this line

<n - Data-shift line left n positions

<<n - Data-shift block of lines left n positions

>n - Data-shift line right n positions

>>n - Data-shift block of lines right n positions

(n - Column-shift line left n positions

((n - Column-shift block of lines left n positions

)n - Column-shift line right n positions

))n - Column-shift block of lines right n positions

If a number is not entered, the default is 1 except for:

- The shift commands (which default to 2 column positions)
- The TE command (defaults to the number of lines remaining on the screen)
- The TF command (defaults to the current right boundary)

## DIALOG TEST PRIMARY COMMANDS

The following primary commands have special meaning during dialog test operations. They are entered in the command line of the applicable dialog test option panels.

- CANCEL (CAN)** Ends the current option. Any changes made to the data are ignored.
- END** Ends the current option. Any changes made to the data now take effect.
- LOCATE (LOC, L) argument** Positions the scrollable display to the first (next) row containing the argument. The scan starts at the end of the first row currently being displayed. A message is displayed indicating the result of the scan.
- QUAL** This command is entered only from the breakpoints panel. It displays to the user the breakpoint qualification data.
- RESUME (RES)** This command is entered on the qualification panel. It restores the format of the breakpoints panel. Each breakpoint that has qualification is flagged by the characters "\*QUAL\*" in columns 75 to 80 on that line of the breakpoints panel.

## DIALOG TEST LINE COMMANDS

The following line commands have special meaning during testing operations:

- D or Dn** Delete one line or n lines starting with this line.
- I or In** Insert one line or n lines directly after this line (with underscores and quotes in the appropriate fields).
- R or Rn** Repeat this line once or n times.

## APPENDIX 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

CAPS [ON ]  
[OFF]

COLS

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

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

LOCATE line-number  
label

.xxxxxx

RESET

### EDIT PRIMARY COMMANDS

AUTOLIST [ON ]



```

                [OFF]

AUTONUM  [ON ]
         [OFF]

CANCEL
CAN

CAPS    [ON ]
        [OFF]

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

COPY    [member-name]

CREATE  [member-name]

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

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

LOCATE  line-number

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

MOVE   [member-name]

NULLS  [ON ] [ALL]
       [OFF]

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

PROFILE [name] [number]

RECOVERY [ON ]
        [OFF]

RENUM  [STD] [COBOL] [DISPLAY]

REPLACE [member-name]

RESET

```

SAVE

STATS [ON ]  
[OFF]

SUBMIT

TABS [ON ] [tab-char] [ALL]  
[OFF]

UNNUM

## FIND/CHANGE STRINGS (BROWSE AND EDIT)

### Character String Format:

|                   |         |                                |
|-------------------|---------|--------------------------------|
| Simple string:    | cccccc  | (no embedded blanks or commas) |
| Delimited string: | 'cccc'  | or "cccc"                      |
| Hex string:       | X'hhhh' | or 'hhhh'X                     |
| Text string:      | T'cccc' | or 'cccc'T                     |
| Picture string:   | P'ssss' | or 'ssss'P                     |
| Previous string:  | *       | (single asterisk)              |

### Picture Strings - Special Characters:

P'=' - any character (don't care)  
P'-' - any nonblank character  
P'.' - any non-displayable (invalid) character  
P' #' - any numeric character (0-9)  
P'-' - any non-numeric character  
P'@' - any alphabetic character (uppercase or lowercase)  
P'<' - any lowercase alphabetic character  
P'>' - any uppercase alphabetic character  
P'\$' - any special character (not alphabetic or numeric)

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

P'Ⓢ' - any APL-specific or TEXT-specific character

## EDIT LINE COMMANDS

|                 |              |
|-----------------|--------------|
| A, An           | (after)      |
| B, Bn           | (before)     |
| BOUNDS          |              |
| COLS            |              |
| C, Cn, CC       | (copy)       |
| D, Dn, DD       | (delete)     |
| F, Fn           | (first)      |
| I, In           | (insert)     |
| L, Ln           | (last)       |
| MASK            |              |
| M, Mn, MM       | (move)       |
| O, On, OO       | (overlay)    |
| R, Rn, RR, RRn  | (repeat)     |
| S, Sn           | (show)       |
| TABS            |              |
| TE, TEn         | (text entry) |
| TF, TFn         | (text flow)  |
| TS, TSn         | (text split) |
| X, Xn, XX       | (exclude)    |
| <, <n, <<, <<n  | (data left)  |
| >, >n, >>, >>n  | (data right) |
| (, (n, ((, ((n  | (cols left)  |
| ), )n, ))), ))n | (cols right) |

## APPENDIX 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

page 121

| Model Name                       | Qualifier | Description                 | Member Name |
|----------------------------------|-----------|-----------------------------|-------------|
| <b>Table Functions (General)</b> |           |                             |             |
| 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     |
| <b>Table Functions (Row)</b>     |           |                             |             |
| 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    |
| TBTOP                            |           | Table top                   | ISREMC9     |
| TBVCLER                          |           | Table variable clear        | ISREMCRC    |
| <b>Variables</b>                 |           |                             |             |
| VGET                             |           | Variable get                | ISREMCV1    |
| VPUT                             |           | Variable put                | ISREMCV2    |

| Model Name | Qualifier | Description | Member Name |
|------------|-----------|-------------|-------------|
|------------|-----------|-------------|-------------|

### File Tailoring

|         |  |                              |          |
|---------|--|------------------------------|----------|
| FTCLOSE |  | End file tailoring           | ISREDCF3 |
| FTERASE |  | File tailor erase            | ISREDCF4 |
| FTINCL  |  | File tailor include skeleton | ISREDCF2 |
| FTOPEN  |  | File tailor open             | IREMDCF1 |

### Display

|         |  |                           |          |
|---------|--|---------------------------|----------|
| DISPLAY |  | Display option            | ISREDCD1 |
| SETMSG  |  | Set message display       | ISREDCD3 |
| TBDISPL |  | Table display information | ISREDCD2 |

### Miscellaneous

|         |  |                           |          |
|---------|--|---------------------------|----------|
| BROWSE  |  | Browse service            | ISREDCM3 |
| CONTROL |  | Control service           | ISREDCM2 |
| EDIT    |  | Edit service              | ISREDCM4 |
| LOG     |  | Write message or log file | ISREDCM5 |
| SELECT  |  | Select service            | ISREDCM1 |

## ISPF SERVICES IN COBOL PROGRAMS

### Working Storage Section

|          |  |                            |          |
|----------|--|----------------------------|----------|
| WORKSTOR |  | Working storage definition | ISREMBW1 |
|----------|--|----------------------------|----------|

### Table Functions (General)

|          |  |              |          |
|----------|--|--------------|----------|
| TBCLOSE  |  | Table close  | ISREMBG5 |
| TBCREATE |  | Table create | ISREMBG1 |
| TBEND    |  | Table end    | ISREMBG6 |
| TBERASE  |  | Table erase  | ISREMBG7 |
| TBOPEN   |  | Table open   | ISREMBG2 |
| TBQUERY  |  | Table query  | ISREMBG3 |
| TBSAVE   |  | Table save   | ISREMBG4 |

### Table Functions (Row)

|          |  |                             |          |
|----------|--|-----------------------------|----------|
| TBADD    |  | Table row add               | ISREMBR1 |
| TBBOTTOM |  | Table row pointer to bottom | ISREMBRA |
| TBDELETE |  | Table delete                | ISREMBR2 |
| TBEXIST  |  | Table exist                 | ISREMBR6 |
| TBGET    |  | Table get                   | ISREMBR3 |
| TBPUT    |  | Table put                   | ISREMBR4 |
| TBMOD    |  | Table modify                | ISREMBR5 |
| TBSARG   |  | Table search argument       | ISREMBR7 |
| TBSCAN   |  | Table scan                  | ISREMBR8 |
| TBSKIP   |  | Table skip                  | ISREMBRB |
| TBTOP    |  | Table top                   | ISREMBR9 |

| Model Name            | Qualifier | Description                  | Member Name |
|-----------------------|-----------|------------------------------|-------------|
| TBVCLEAR              |           | Table variable clear         | ISREMBRC    |
| <b>Variables</b>      |           |                              |             |
| 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    |
| <b>File Tailoring</b> |           |                              |             |
| FTCLOSE               |           | End file tailoring           | ISREMBF3    |
| FTERASE               |           | File tailor erase            | ISREMBF4    |
| FTINCL                |           | File tailor include skeleton | ISREMBF2    |
| FTOPEN                |           | File tailor open             | IREMMBF1    |
| <b>Display</b>        |           |                              |             |
| DISPLAY               |           | Display option               | ISREMBD1    |
| SETMSG                |           | Set message display          | ISREMBD3    |
| TBDISPL               |           | Table display information    | ISREMBD2    |
| <b>Miscellaneous</b>  |           |                              |             |
| BROWSE                | MVS       | Browse service (MVS)         | ISREMCB1    |
| BROWSE                | VM        | Browse service (VM)          | ISREMCB2    |
| CONTROL               |           | Control service              | ISREMBM2    |
| EDIT                  | MVS       | Edit service (MVS)           | ISREMCCE1   |
| EDIT                  | VM        | Edit service (VM)            | ISREMCCE2   |
| LOG                   |           | Write message or log file    | ISREMBM5    |
| SELECT                |           | Select service               | ISREMBM1    |

## ISPF SERVICES IN EXEC

| 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 |

### Table Functions (Row)

|          |  |                             |          |
|----------|--|-----------------------------|----------|
| TBADD    |  | Table row add               | ISREMER1 |
| TBBOTTOM |  | Table row pointer to bottom | ISREMER8 |
| 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                  | ISREMER8 |
| TBTOP    |  | Table top                   | ISREMER9 |
| TBVCLER  |  | Table variable clear        | ISREMER9 |

### Variables

|      |  |              |          |
|------|--|--------------|----------|
| VGET |  | Variable get | ISREMEV1 |
| VPUT |  | Variable put | ISREMEV2 |

### File Tailoring

|         |  |                              |          |
|---------|--|------------------------------|----------|
| FTCLOSE |  | End file tailoring           | ISREMEF3 |
| FTERASE |  | File tailor erase            | ISREMEF4 |
| FTINCL  |  | File tailor include skeleton | ISREMEF2 |
| FTOPEN  |  | File tailor open             | IREMMCF1 |

### Display

|         |  |                           |          |
|---------|--|---------------------------|----------|
| DISPLAY |  | Display option            | ISREMED1 |
| SETMSG  |  | Set message display       | ISREMED3 |
| TBDISPL |  | Table display information | ISREMED2 |

| Model Name                               | Qualifier | Description                 | Member Name |
|--|-----------|-----------------------------|-------------|
| <b>Miscellaneous</b>                     |           |                             |             |
| BROWSE                                   |           | Browse service              | ISREMEM3    |
| CONTROL                                  |           | Control service             | ISREMEM2    |
| EDIT                                     |           | Edit service                | ISREMEM4    |
| LOG                                      |           | Write message or log file   | ISREMEM5    |
| SELECT                                   |           | Select service              | ISREMEM1    |
| <b>ISPF SERVICES IN FORTRAN PROGRAMS</b> |           |                             |             |
| <b>Table Functions (General)</b>         |           |                             |             |
| TBCLOSE                                  |           | Table close                 | ISREMF5     |
| TBCREATE                                 |           | Table create                | ISREMF1     |
| TBEND                                    |           | Table end                   | ISREMF6     |
| TBERASE                                  |           | Table erase                 | ISREMF7     |
| TBOPEN                                   |           | Table open                  | ISREMF2     |
| TBQUERY                                  |           | Table query                 | ISREMF3     |
| TBSAVE                                   |           | Table save                  | ISREMF4     |
| <b>Table Functions (Row)</b>             |           |                             |             |
| TBADD                                    |           | Table row add               | ISREMF1     |
| TBBOTTOM                                 |           | Table row pointer to bottom | ISREMFRA    |
| TBDELETE                                 |           | Table delete                | ISREMF2     |
| TBEXIST                                  |           | Table exist                 | ISREMF6     |
| TBGET                                    |           | Table get                   | ISREMF3     |
| TBMOD                                    |           | Table modify                | ISREMF5     |
| TBPUT                                    |           | Table put                   | ISREMF4     |
| TBSARG                                   |           | Table search argument       | ISREMF7     |
| TBSCAN                                   |           | Table scan                  | ISREMF8     |
| TBSKIP                                   |           | Table skip                  | ISREMF8B    |
| TBTOP                                    |           | Table top                   | ISREMF9     |
| TBVCLER                                  |           | Table variable clear        | ISREMFRC    |
| <b>Variables</b>                         |           |                             |             |
| VCOPY                                    |           | Copy variable               | ISREMFV5    |
| VDEFINE                                  |           | Variable define             | ISREMFV3    |
| VDELETE                                  |           | Variable delete             | ISREMFV4    |
| VGET                                     |           | Variable get                | ISREMFV1    |
| VPUT                                     |           | Variable put                | ISREMFV2    |
| VREPLACE                                 |           | Variable replace            | ISREMFV6    |
| VRESET                                   |           | Variable reset              | ISREMFV7    |



| Model Name                          | Qualifier | Description                         | Member Name |
|-------------------------------------|-----------|-------------------------------------|-------------|
| <b>File Tailoring</b>               |           |                                     |             |
| FTCLOSE                             |           | End file tailoring                  | ISREFFF3    |
| FTERASE                             |           | File tailor erase                   | ISREFFF4    |
| FTINCL                              |           | File tailor include skeleton        | ISREFFF2    |
| FTOPEN                              |           | File tailor open                    | IREMMFF1    |
| <b>Display</b>                      |           |                                     |             |
| DISPLAY                             |           | Display option                      | ISREFFD1    |
| SETMSG                              |           | Set message display                 | ISREFFD3    |
| TBDISPL                             |           | Table display information           | ISREFFD2    |
| <b>Miscellaneous</b>                |           |                                     |             |
| BROWSE                              | MVS       | Browse service (MVS)                | ISREFFB1    |
| BROWSE                              | VM        | Browse service (VM)                 | ISREFFB2    |
| CONTROL                             |           | Control service                     | ISREFFM2    |
| EDIT                                | MVS       | Edit service (MVS)                  | ISREFFE1    |
| EDIT                                | VM        | Edit service (VM)                   | ISREFFE2    |
| LOG                                 |           | Write message or log file           | ISREFFM5    |
| SELECT                              |           | Select service                      | ISREFFM1    |
| <b>MESSAGE FORMAT</b>               |           |                                     |             |
| MSG                                 |           | Message member selection            | ISREFFMSG   |
| <b>PANEL FORMATS AND STATEMENTS</b> |           |                                     |             |
| <b>Panel Formats</b>                |           |                                     |             |
| ENTRY                               |           | Data entry panel                    | ISREMMF1    |
| MULTIPLE                            |           | Multiple-column panel               | ISREMSSEL   |
| MULTIPLE                            | SELECT2   | Double-column selection panel       | ISREMSSE1   |
| MULTIPLE                            | ENTRY2    | Double-column entry panel           | ISREMSSE2   |
| SELECTION                           |           | Menu/selection panel                | ISREMMF3    |
| TBDISPL                             |           | Table display table                 | ISREMMF4    |
| TUTORIAL                            |           | Help/tutorial panel                 | ISREMMF5    |
| <b>Panel Statements</b>             |           |                                     |             |
| ASSIGN                              |           | Assignment statement                | ISREMASG    |
| ASSIGN                              | SIMPLE    | Simple assignment statement         | ISREMAS1    |
| ASSIGN                              | TRANS     | Trans assignment statement          | ISREMAS2    |
| ASSIGN                              | TRUNC     | Trunc assignment statement          | ISREMAS3    |
| ASSIGN                              | TRANSTRU  | Nested translate truncate statement | ISREMAS4    |
| ATTR                                |           | Attribute section header            | ISREMS2     |
| ATTRIB                              |           | New attribute character definition  | ISREMS3     |
| BODY                                |           | Body section header                 | ISREMS4     |
| CONTROL                             |           | Control variables                   | ISREMCNL    |

| Model Name | Qualifier | Description                           | Member Name |
|------------|-----------|---------------------------------------|-------------|
| CONTROL    | CURSOR    | Control initial cursor placement      | ISREMCN1    |
| CONTROL    | HELP      | Establish a tutorial panel            | ISREMCN2    |
| CONTROL    | MSG       | Identify message to be displayed      | ISREMCN3    |
| CONTROL    | RESP      | Indicate user response to panel       | ISREMCN4    |
| CONTROL    | TRAIL     | Contain remainder from TRUNC function | ISREMCN5    |
| IF         |           | IF statement                          | ISREMMS6    |
| MODEL      |           | Model section header                  | ISREMMS7    |
| VERIFY     |           | Verification statements               | ISREMVÉR    |
| VERIFY     | ALPHA     | Alphabetic characters                 | ISREMVÉ1    |
| VERIFY     | BIT       | Binary characters                     | ISREMVÉ2    |
| VERIFY     | DSNAME    | TSO data set name                     | ISREMVÉ3    |
| VERIFY     | FILEID    | CMS fileid                            | ISREMVÉ4    |
| VERIFY     | HEX       | Hexadecimal characters                | ISREMVÉ5    |
| VERIFY     | LIST      | List of valid values                  | ISREMVÉ6    |
| VERIFY     | NAME      | Data set member name                  | ISREMVÉ7    |
| VERIFY     | NONBLANK  | Verify nonblank field                 | ISREMVÉ8    |
| VERIFY     | NUM       | Numeric characters                    | ISREMVÉ9    |
| VERIFY     | PICT      | Mixed characters matching picture     | ISREMVÉA    |
| VERIFY     | RANGE     | Numeric value within specified limits | ISREMVÉB    |
| VPUT       |           | Variable put statement                | ISREMMS9    |

## ISPF SERVICES IN PL/I PROGRAMS

### Table Functions (General)

|          |              |          |
|----------|--------------|----------|
| TBCLOSE  | Table close  | ISREMPG5 |
| TBCREATE | Table create | ISREMPG1 |
| TBEND    | Table end    | ISREMPG6 |
| TBERASE  | Table erase  | ISREMPG7 |
| TBOPEN   | Table open   | ISREMPG2 |
| TBQUERY  | Table query  | ISREMPG3 |
| TBSAVE   | Table save   | ISREMPG4 |

### 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 |

| Model Name                               | Qualifier | Description                       | Member Name |
|--|-----------|-----------------------------------|-------------|
| TBSARG                                   |           | Table search argument             | ISREMPR7    |
| TBSCAN                                   |           | Table scan                        | ISREMPR8    |
| TBSKIP                                   |           | Table skip                        | ISREMPRB    |
| TBTOP                                    |           | Table top                         | ISREMPR9    |
| TBVCLEAR                                 |           | Table variable clear              | ISREMPRC    |
| <b>Display</b>                           |           |                                   |             |
| DISPLAY                                  |           | Display option                    | ISREMPD1    |
| SETMSG                                   |           | Set message display               | ISREMPD3    |
| TBDISPL                                  |           | Table display information         | ISREMPD2    |
| <b>Variables</b>                         |           |                                   |             |
| 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    |
| <b>Miscellaneous</b>                     |           |                                   |             |
| BROWSE                                   | MVS       | Browse service (MVS)              | ISREMPB1    |
| BROWSE                                   | VM        | Browse service (VM)               | ISREMPB2    |
| CONTROL                                  |           | Control service                   | ISREMPM2    |
| EDIT                                     | MVS       | Edit service (MVS)                | ISREMPE1    |
| EDIT                                     | VM        | Edit service (VM)                 | ISREMPE2    |
| LOG                                      |           | Write message or log file         | ISREMPM5    |
| SELECT                                   |           | SELECT service                    | ISREMPM1    |
| <b>FILE TAILORING CONTROL STATEMENTS</b> |           |                                   |             |
| CM                                       |           | Define comment statement          | ISREMSK1    |
| DEFAULT                                  |           | Change control character defaults | ISREMSK2    |
| DOT                                      |           | Define DO group for table row     | ISREMSK3    |
| IM                                       |           | Imbed specified file skeleton     | ISREMSK4    |
| SEL                                      |           | Conditional execution definition  | ISREMSK5    |
| SET                                      |           | Set dialog variable value         | ISREMSK6    |
| TB                                       |           | Set tab stop position             | ISREMSK7    |

## INDEX

### Special Characters

- < (shift data left) line command
  - edit 113
- ( (shift columns left) line command
  - edit 113
- ) (shift columns right) line command
  - edit 113
- > (shift data right) line command
  - edit 113

### A

- A (after) line command 95
- add after row entry panel
  - dialog test 203
- add after row tables option
  - dialog test 202
- adding dialog variables
  - dialog test 191
- adding lines 91
- adding models 126
- allocating a new data set (data set utility) 137
- APL keyboard
  - internal character translation 241
- application command table 157
- assembler/compiler processing 165
- ATTENTION (PA1) key 17
- AUTOLIST command 66
- autolist mode 66
  - edit 58
- automatic source listing
  - edit 66
- AUTONUM command
  - edit 65
- autonum mode
  - edit 58, 65

### B

- B (before) line command 95
- batch (option 5) 175
- batch output listing
  - allocation 36
  - browsing 36
  - printing 36
- bottom
  - scrolling to 18
- boundary definition line
  - edit 112
- BOUNDS line command
  - edit 111
- breakpoint primary option menu
  - dialog test 222
- breakpoint qualification panel
  - dialog test 220
- breakpoints entry panel
  - dialog test 219
- breakpoints option
  - dialog test 217
- browse (option 1) 45
  - CAPS command 51
  - column identification line 48
  - COLUMNS command 48
  - command summary 245
  - commands
    - syntax 251
  - data set characteristics 45
  - FIND command 49
  - find/change strings 253
  - functions 47
  - HEX command 51
  - label 47
  - LOCATE command 47
  - program function keys 17
  - RESET command 48
  - scale line 48
  - setting caps mode 51
  - setting hex mode 51
- browse log option
  - dialog test 208
- browsing a member (library utility) 135

## C

- C (copy) line command
  - used with CREATE or REPLACE 86
- C (copy) line command 96
- CANCEL command
  - dialog test 186
  - edit 89
- CAPS command
  - browse 51
  - edit 66
- caps mode
  - edit 59, 66
- catalog management utility (option 3.4) 147
- cataloging a data set (data set utility) 139
- caution message
  - edit 61
- chaining commands
  - See stacking commands
- CHANGE command
  - edit 75
- CHANGE PF key function
  - See RCHANGE command
- changing
  - indentations
    - edit 113
- changing data
  - edit 75
- changing dialog variables
  - dialog test 191
- changing models 126
- changing PF keys
  - KEYS command 15
- changing program function keys 41
- character translation
  - edit 66
- CLIST
  - ISPF services in 255
- closed table status display panel
  - dialog test 206
- COBOL
  - programs
    - ISPF services in 256
- column identification line
  - browse 48
  - edit 106
- COLUMNS
  - line command
    - edit 106
  - primary command
    - browse 48
  - shifting 113
- command (option 6) 179
  - restrictions 179
- command table
  - application 157
- command table utility (option 3.9) 157
  - line commands 158
  - primary commands 159
- command/option area 6
- commands
  - browse (option 1)
    - summary 245
    - syntax 251
  - command table utility (option 3.9)
    - line 158
    - primary 159
  - description summary 243
  - dialog test line 185, 187
    - summary 250
  - dialog test primary 185, 186
    - summary 250
  - edit (option 2) line
    - summary 247
    - syntax 254
  - edit (option 2) primary
    - summary 245
    - syntax 251
  - entering 13
  - general
    - description 14
    - PF keys assigned 14
    - summary 243
  - member list
    - summary 244
    - syntax 251
  - scrolling 18
  - stacking 13, 39
  - syntax
    - summary 251
- compressing a data set (library utility) 135
- concatenating libraries 25
- controlling nulls
  - edit 67
- convert menus/messages utility (option 3.9) 159
- COPY command (edit) 83
- CREATE command
  - edit 85
- creating a new member 53
- creating or updating statistics (reset statistics utility) 148
- CURSOR command 14

## D

- D (delete) line command
    - dialog test 187
    - edit 92
  - data
    - shifting 113
  - data set characteristics
    - browse 45
    - edit 53
  - data set password specification 31
  - data set security 31
  - data set utility (option 3.2) 137
  - default edit mode settings 60
  - defining boundary columns
    - edit 111
  - defining masks
    - edit 107
  - defining program function keys 41
    - KEYS command 41
    - specifying line commands 43
    - specifying primary commands 42
  - delete row tables option
    - dialog test 199
  - delete table row confirmation panel
    - dialog test 199
  - deleting a catalog alias (catalog utility) 147
  - deleting a member (library utility) 135
  - deleting held job output (outlist utility) 155
  - deleting lines 92
  - deleting models 126
  - deleting statistics (reset statistics utility) 149
  - delimiter
    - command stacking 13, 39
  - dialog development 2
    - models 121, 255
  - dialog test 1
  - dialog test (option 7) 181
    - breakpoints option 217
    - browse log option 208
    - environment 182
    - functions option 187
    - invoke service option 211
    - line commands 185, 187
      - summary 250
    - panels option 189
    - primary commands 185, 186
      - summary 250
    - primary option menu 182
    - severe error handling 184
    - tables option 195
      - add after row 202
      - delete row 199
      - display row 197
      - display status 205
      - display structure 204
      - modify row 200
    - traces option 212
    - variable usage 183
    - variables option 191
  - dialog variables
    - manipulating 193
  - display panel entry panel
    - dialog test 189
  - display row tables option
    - dialog test 197
  - display status tables option
    - dialog test 205
  - display structure tables option
    - dialog test 204
  - display table row entry panel
    - dialog test 197
  - displaying a member list (library utility) 135
  - displaying a message 190
  - displaying a panel 189
  - displaying catalog entries (catalog utility) 147
  - displaying data in hexadecimal format
    - browse 51
  - displaying data set information (data set utility) 139
  - displaying dialog variables
    - dialog test 191
  - displaying held job output (outlist utility) 156
  - displaying the panel identifier
    - PANELID command 15
  - documentation preparation 1
  - DOWN command 14
  - draft document options
    - SCRIPT/VS 171
  - DSPRINT TSO command processor 152
  - duplicating lines 93
- ## E
- edit (option 2) 53
    - < (shift data left) line command 113
    - ( (shift columns left) line command 113

) (shift columns right) line command 113

> (shift data right) line command 113

A (after) line command 95

AUTOLIST command 66

autolist mode 58, 66

AUTONUM command 65

autonum mode 58, 65

B (before) line command 95

boundary definition line 112

BOUNDS line command 111

C (copy) line command 96

CANCEL command 89

CAPS command 66

caps mode 59, 66

caution message 61

CHANGE command 75

changing indentations 113

column identification line 106

COLUMNS line command 106

COPY command 83

CREATE command 85

D (delete) line command 92

data set characteristics 53

default mode settings 60

defining boundary columns 111

defining masks 107

excluding lines 116

F (first) line command 119

FIND command 75

find/change strings 253

functions 57

HEX command 72

hex mode 59, 72

hexadecimal display 71

I (insert) line command 91

identifying columns 106

L (last) line command 119

line command summary 247

line commands 58, 89

    syntax 254

LOCATE command 74

locating lines 74

M (move) line command 95

MASK line command 107

merging data 83

MODEL command 58, 123

models 121

modes 58

MOVE command 83

NULLS command 67

nulls mode 59, 67

NUMBER command 64

number mode 58, 64

O (overlay) line command 104

overlying lines 104

preparing text 97

primary command summary 245

primary commands 57

    syntax 251

profile 59

PROFILE command 70

profile display 70

program function keys 17

R (repeat) line command 93

RECOVERY command 73

recovery control 73

recovery mode 59, 73

recovery panel 73

RENUM command 65

REPLACE command 85

RESET command 61, 70, 73

S (show) line command 119

SAVE command 88

scale line 106

segmenting data 83

sequence number display 63

sequence number format 62

sequence number generation and control 62

STATS command 66

stats mode 58, 66

SUBMIT command 87

tab definition line 110

tabs control 68, 109

tabs definition 68

TABS line command 109

tabs mode 59, 69

TABS primary command 69

TE (text entry) line command 98

termination 61

TF (text flow) line command 102

TS (text split) line command 99

UNNUM command 65

X (exclude) line command 117

edit modes 58

edit profile 59

    display and control 70

    specifying 55

edit recovery panel 73

editing a new member 53

encountering a breakpoint

    dialog test 222

END command 14, 22

    dialog test 186

entering commands 13

entry panel 6

excluding lines  
     edit 116  
 EXECs  
     ISPF services in 258  
 extended return  
     See jump function

**F**

F (first) line command  
     edit 119  
 file tailoring control statements  
     models 262  
 final document options  
     SCRIPT/VS 173  
 FIND command  
     browse 49  
     edit 75  
 FIND PF key function  
     See RFIND command  
 find/change strings  
     browse (option 1) 253  
     edit (option 2) 253  
     specification 79  
 finding character strings  
     browse 49  
 finding data  
     edit 75  
 foreground (option 4) 163  
     processing sequence 164  
     SCRIPT/VS processing 170  
 foreground output listing  
     allocation 36  
     browsing 36  
     printing 36  
 format of panels 5  
 FORTRAN  
     programs  
         ISPF services in 259  
 function trace entry panel  
     dialog test 214  
 functions option  
     dialog test 187

**G**

general commands  
     summary 243  
 generating JCL for printing or  
     punching (hardcopy utility) 151  
 getting message explanation 21

**H**

hardcopy utility (option 3.6) 150  
 hardware tabs (edit) 68  
 HELP command 14, 21  
     tutorial (option T) 225  
 help information 21  
 HEX command  
     browse 51  
     edit 72  
 hex mode  
     browse 51  
     edit 59, 72  
 hexadecimal display  
     edit 71

**I**

I (insert) line command  
     dialog test 187  
     edit 91  
 identifying columns  
     browse 48  
     edit 106  
 indentations  
     changing  
         edit 113  
 index listing  
     format 235  
 initializing a user catalog alias  
     (catalog utility) 147  
 inserting lines 91  
 invocation of PDF 10  
 invoke dialog service entry panel  
     dialog test 211  
 invoke function entry panel  
     dialog test 188  
 invoke service option  
     dialog test 211  
 invoking  
     command procedure 187  
     panel 187  
     program 187  
 ISPF library 25  
 ISPF services  
     in CLIST 255  
     in COBOL programs 256  
     in EXECs 258  
     in FORTRAN programs 259  
     in PL/I programs 261



## J

- JCL for printing or punching
  - generating (hardcopy utility) 151
- job statement specification 31
- job statements
  - in user profile 32
- jump function 15, 22

## K

- KEYS command 15

## L

- L (last) line command
  - edit 119
- label
  - browse 47
- language processors 1
- LEFT command 14
- library and data set name specification 29
- library concatenation 25
- library hierarchy 25
- library statistics 27
  - reset statistics utility 28
- library utility (option 3.1) 134
- line commands
  - command table utility (option 3.9) 158
  - edit 89
    - rules for entering 90
- linkage editor processing 168
- list and log files
  - format 235
- list file
  - description 35
- list VTOC utility (option 3.7) 152
- listing job names (outlist utility) 155
- LOCATE command
  - dialog test 186
  - edit 74
  - member list 33
  - under browse 47
- locating lines
  - browse 47
  - edit 74
- locating members 33
- locating models 132

- log and list files 35
  - allocation 35
  - printing 35
  - setting the default allocation 40
  - specifying defaults 39
- log file
  - description 35
- log listing
  - format 236
- logical screens
  - switching between 20
- logical tabs (edit) 68
- long message area 6

## M

- M (move) line command 95
  - used with CREATE or REPLACE 86
- manipulating dialog variables 193
- MASK line command
  - edit 107
- member list 6, 33
  - browsing a member 34
  - commands 33
    - summary 244
    - syntax 251
  - deleting a member 34
  - LOCATE command 33
  - multiple-selection 34
  - printing a member 34
  - renaming a member 34
  - scrolling 33
  - SELECT command 33
  - selecting a member 34
  - selection codes 33, 34
  - single-selection 33
- menu
  - See selection panel
- merging data (edit) 83
- message
  - long 6
  - short 5
- message display panel
  - dialog test 190
- message explanation 21
- message format models 260
- messages
  - displaying 190
- model class 122
- MODEL command
  - edit 123
- model name 122

- model name qualifier 122
- model selection panel 125
- models 1, 2, 121, 255
  - adding 126
  - changing 126
  - class 122
  - deleting 126
  - dialog development 255
  - file tailoring control
    - statements 262
  - hierarchy 122
  - locating 132
  - logical name 122
  - message format 260
  - name 122
  - name qualifier 122
  - panel format and statement 260
  - types 122
  - using 124
- modify row tables option
  - dialog test 200
- modify table row entry panel
  - dialog test 201
- MOVE command (edit) 83
- move/copy utility (option 3.3) 141

## N

- nondisplayable characters
  - edit 53
- nondisplayable data
  - dialog test 194, 201, 204, 211
- NULLS command
  - edit 67
- nulls mode
  - edit 59, 67
- NUMBER command
  - edit 64
- number mode
  - edit 58, 64

## O

- O (overlay) line command
  - edit 104
- online tutorial 1
- open table status display panel
  - dialog test 207
- option X 22
- outlist utility (option 3.8) 154
- overlying lines
  - edit 104,

- overtyping of parameters 19

## P

- pad character 39
- panel 5
  - add after row entry
    - dialog test 203
  - breakpoint primary option menu
    - dialog test 222
  - breakpoint qualification
    - dialog test 220
  - breakpoints entry
    - dialog test 219
  - closed table status display
    - dialog test 206
  - command/option area 6
  - delete table row confirmation
    - dialog test 199
  - dialog test primary option
    - menu 182
  - display panel entry
    - dialog test 189
  - display table row entry
    - dialog test 197
  - edit recovery 73
  - format 5
    - 3278 Model 5 5
  - format and statement models 260
  - function trace entry
    - dialog test 214
  - input field padding 39
  - invoke dialog service entry
    - dialog test 211
  - invoke function entry
    - dialog test 188
  - invoking
    - dialog test 187
  - long message area 6
  - message display
    - dialog test 190
  - model selection 125
  - modify table row entry
    - dialog test 201
  - open table status display
    - dialog test 207
  - parameter selection 37
  - primary option menu 10
  - scroll area 6
  - short message area 5
  - special display
    - dialog test 212
  - table structure display

- dialog test 205
- tables selection
  - dialog test 195
- terminal characteristics 37
- termination 22
- title area 5
- traces selection
  - dialog test 213
- types 6
- variable trace entry
  - dialog test 216
- variables entry
  - dialog test 192
- PANELID command 15
- panels
  - displaying 189
- panels option
  - dialog test 189
- parameter selection panel 37
- parameter specification 29
- Parms option (option 0) 37
- password protection 31
- PDF 1
- PDF primary options 11
- PF keys
  - See program function keys
- picture string (edit FIND or CHANGE) 81
- PL/I
  - programs
    - ISPF services in 261
- preparing text
  - edit 97
- primary commands
  - command table utility (option 3.9) 159
- primary option menu 10
- PRINT command 15
- print mode
  - See autolist mode
- PRINT-HI command 15
- printer control characters
  - edit 53
- printing a member (library utility) 135
- printing a screen image 15
- printing an entire data set (library utility) 135
- printing an index listing (library utility) 135
- printing catalog entries (catalog utility) 147
- printing held job output
  - outlist utility 155

- printing VTOC entries (list VTOC utility) 152
- PROFILE command
  - edit 70
- profile display
  - edit 70
- program access keys 17
  - ATTENTION (PA1) key 17
  - RESHOW (PA2) key 17
- program function keys 13, 17
  - changing 41
  - defining 41
  - specifying 38, 41

## Q

- QUAL command
  - dialog test 186
- qualifying a breakpoint
  - dialog test 219
- qx 132

## R

- R (repeat) line command
  - dialog test 187
  - edit 93
- range and column limitation (edit FIND and CHANGE) 82
- RCHANGE command 14
- RCHANGE PF key
  - edit 76
- RECOVERY command
  - edit 73
- recovery control
  - edit 73
- recovery mode
  - edit 59, 73
- renaming a data set (data set utility) 138
- renaming a member (library utility) 135
- RENUM command
  - edit 65
- repeating lines 93
- REPLACE command
  - edit 85
- requeuing held job output (outlist utility) 156
- RESET command
  - browse 48
  - edit 61, 70, 73

- reset statistics utility (option 3.5) 148
- RESHOW (PA2) key 17
- RESUME command
  - dialog test 186
- RETURN command 14, 22
- RFIND command 14
  - with FIND command (browse) 50
- RFIND PF key
  - edit 76
- RIGHT command 14
  
- S**
- S (show) line command
  - edit 119
- SAVE command
  - edit 88
- saving data
  - edit 88
- scale line
  - See column identification line
- SCRIPT/VS processing
  - draft document options 171
  - final document options 173
  - foreground 170
- scroll area 6
- scrollable data display 6
- scrolling 1, 18
  - commands 18
  - DOWN command 14
  - during browse 46
  - LEFT command 14
  - member list 33
  - RIGHT command 14
  - scroll amount 18
  - to bottom 18
  - to top 18
  - UP command 14
  - use of cursor 19
- scrolling members 33
- security
  - data set 31
- segmenting data (edit) 83
- SELECT command
  - member list 33
- selection codes
  - member list 33
- selection panel 6
- separator
  - command stacking 13, 39
- sequence number
  - display 63
  - format 62
  - generation and control 62
- Session Manager 163, 180
- setting caps mode
  - browse 51
- shifting data 113
- short message area 5
- software tabs (edit) 68
- special display panel
  - dialog test 212
- specifying
  - data set passwords 31
  - job statement information 31
  - library and data set names 29
  - log and list data set
    - disposition at termination 22
  - pad character 39
  - parameters 29
  - PF keys 38
  - program function key
    - definitions 41
  - screen format (3278 Model 5) 39
  - stacking delimiter 39
  - terminal characteristics 37
  - terminal type 38
- specifying log and list file
  - defaults 39
- SPLIT command 14
- split screen 1, 20
  - entering 20
  - terminating 20
  - using the SPLIT command 20
  - using the SPLIT PF key 20
  - using the SWAP command 20
- stacking commands 13, 39
- statistics generation
  - edit 66
- STATS command
  - edit 66
- stats mode
  - edit 58, 66
- structured programming 2
- SUBMIT command (edit) 87
- submitting to a job stream
  - (edit) 87
- summary of command syntax 251
- summary of commands 243
- SWAP command 14
- switching between logical
  - screens 20

## T

- tab definition line
  - edit 110
- table structure display panel
  - dialog test 205
- tables option
  - dialog test 195
- tables selection panel
  - dialog test 195
- tabs control
  - edit 68, 109
- tabs definition
  - edit 68
- TABS line command
  - edit 109
- tabs mode
  - edit 59, 69
- TABS primary command
  - edit 69
- TE (text entry) line command
  - edit 98
- terminal characteristics
  - specifying 37
- terminal characteristics panel 37
- terminal keys 16
- terminal type 38
- termination
  - edit 61
  - options 22
    - END command 22
    - jump function 22
    - option X 22
    - RETURN command 22
  - panel 22
  - PDF 22
  - specifying log and list file defaults 39
- testing dialogs 181
- TEXT keyboard
  - internal character translation 241
- text preparation
  - edit 97
- text strings (edit FIND or CHANGE) 81
- TF (text flow) line command
  - edit 102
- title area of panel 5
- top
  - scrolling to 18

- trace output 209
  - function trace entries 209
  - trace header 209
  - variable trace entries 210
- traces option
  - dialog test 212
- traces selection panel
  - dialog test 213
- translating character strings
  - browse 51
- TS (text split) line command
  - edit 99
- TSO command 15
- tutorial (option T) 225
  - HELP command 225
- types of panels 6

## U

- uncataloging a data set (data set utility) 139
- UNNUM command
  - edit 65
- UP command 14
- usage scenario 3, 229
- user profile 32
  - scroll amount 19
- utilities
  - command table utility (option 3.9) 157
- utilities (option 3) 1, 133

## V

- variable trace entry panel
  - dialog test 216
- variables entry panel
  - dialog test 192
- variables option
  - dialog test 191
  - usage 194

## X

- X (exclude) line command
  - edit 117
- X option
  - used with jump function 16

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

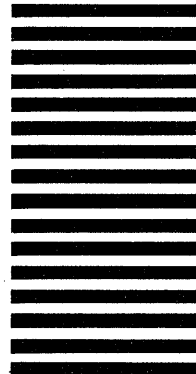
Please Do Not Staple

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 Z59, Building 931  
P. O. Box 390  
Poughkeepsie, New York 12602

Fold and tape

Please Do Not Staple

Fold and tape

ISPF/PDF for MVS: Program Reference

Printed in U.S.A.

SC34-2089-1



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.)



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 Z59, Building 931  
P. O. Box 390  
Poughkeepsie, New York 12602

Fold and tape

Please Do Not Staple

Fold and tape

ISPF/PDF for MVS: Program Reference

Printed in U.S.A.

SC34-2089-1



Publication Number  
SC34-2089-1

File Number  
S370/4300-39

Program Number  
5665-268

Printed in  
USA

**IBM**