



VERSION 7.4

System Services Guide

December 2002

COPYRIGHT

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of SERENA Software, Inc. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by SERENA Software, Inc. SERENA Software, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Copyright © 2002 SERENA Software Inc. All rights reserved.

TRADEMARKS

ChangeMan, Change Man, eChange Man, Comparex, StarTool, StarWarp, UltiMIS, ULTIMIZER, Eye-Spy, Eye-Spy/390, Detect+Resolve, Change Transfer and SERNET are registered trademarks of SERENA Software, Inc. SERENA, the SERENA logo, Merge+Reconcile, FULL.CYCLE, ChangeContent, ChangeXpress, eHealing, eRequestMan, eFull.Cycle, StarProbe, StarSpy, StarTest, StarSuite, and SER10TY are trademarks of SERENA Software, Inc.

IBM, DB2, IMS, ISPF, MVS, MVS/ESA, OS/390, RACF, SAF, and SMF are trademarks and registered trademarks of the International Business Machines Corporation in the United States, other countries, or both. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

Publication Date: December 13, 2002

Order Number: [FDM74IMS](#)

CONTENTS

About This Book

<i>Audience</i>	9
<i>Objectives</i>	9
<i>Manual Organization</i>	10
<i>Related Publications</i>	12
<i>Support</i>	14

Chapter 1: Introduction	15
<i>Data Set Processing</i>	15
<i>New Features</i>	16
Previous Releases Notes	16
<i>Software Requirements</i>	16

Chapter 2: Product Overview	17
<i>StarTool FDM Background</i>	18

Chapter 3: Getting Started	19
<i>Environments</i>	19
<i>ISPMODE Execution</i>	20
Standard Users	20
Advanced Users	25
Power Users	28
<i>Line Mode</i>	29
<i>Batch Mode</i>	31
<i>Started Task Mode</i>	32
<i>Invoking StarTool FDM</i>	32
Single Subcommand Mode	33
License Information	34

Chapter 4:	Menu System	35
	<i>Primary Options Menu</i>	35
	<i>Advanced Options Menu</i>	38
	<i>Navigating in StarTool FDM</i>	40
	<i>Options for Systems Programmers</i>	41
	Option 13 - Examine	41
	Option 13.8 - Current Hardware and Software Environment	43
	Option 13.9 - Customized Defaults	45
Chapter 5:	Dialog Processing	47
	<i>Using CLISTs in ISPMODE</i>	48
	DCONTROL Command Syntax	50
	DSETVAR Command Syntax	53
	Message PDS#102M	53
	DGETVAR Command Syntax	55
	PANEL Command Syntax	56
	CLIST Processing Guidelines	56
Chapter 6:	Batch Processing	65
	<i>Dynamic TSO Environment</i>	68
	<i>ISPMODE in Batch</i>	69
	F Command	72
	EXCLUDE Facility	72
	SORT Command	72
	L (LOCATE)Command	73
	X Command	73
	OUTPUT Command	73
Chapter 7:	Concepts and Facilities	75
	<i>LLA Managed Libraries</i>	76
	LLA and Deleted Members	77
	<i>Using Computer Resources</i>	77
	Paging and Swapping Relief	78

Reducing CPU and I/O	78
Reducing VTOC Contention	79
Chapter 8: Examples	81
Adding Directory Blocks	85
Adding a Single Extent in Tracks	85
Adding a Single Extent in Cylinders	85
Adding an Entry Point	85
Adding ZAP IDR Records	86
Allocating a Data Set	86
Backing Up Multiple Members	86
Building a Member List	87
Changing Secondary Allocation	89
Checking Alias Members	89
Checkpointing the Log	90
Cloning a Member	90
Combining Multiple Members	90
Comparing Members	90
Comparing Multiple Members	91
Compressing Data Sets	91
Compressing Multiple Data Sets	92
Controlling GO Sessions	92
Controlling Multiple Logs	92
Converting CLIST Members	93
Converting Source Members	93
Copying Members	94
Defining a Dynamic Command	94
Deleting All Members	95
Deleting Multiple Members	95
Disassembling a Member	95
Displaying External References	96
Editing Multiple Members	96
Excluding Multiple Members	96
Executing in Background	97

Executing in CLIST Mode	97
Formatting Data Set VTOC Information	98
Formatting Directory Entries	98
Formatting Member Data	98
Listing Module History	99
Locating Data in Dialog Tables	99
Managing Data Set Lists	100
Managing Member Lists	101
Manipulating Non-Partitioned Data Sets	103
Migrating Test Modules to Production	104
Modifying AMODE and RMODE	105
Modifying DCB Parameters	105
Modifying Member Attributes	105
Modifying Member Statistics	106
Modifying Several Members' Attributes	106
Obtaining Assistance:	106
Processing Multiple Data Sets	108
Reblocking Source Members	108
Reconstructing Linkage-Edit JCL	109
Releasing Space in a Data Set	109
Releasing Space in Multiple Data Sets	109
Renaming Multiple Members	110
Repairing a CSECT in Multiple Members	110
Restoring Deleted Members	110
Saving a LISTC Table	111
Saving a MEMLIST Table	111
Searching a Data Set	112
Searching for Panel, Message, or CLIST Members	112
Searching for System Modules	113
Searching Multiple Data Sets	113
Selecting Members	113
Specifying a Member Group	115
Switching Modes	116
Trimming Dialog Tables	117
Updating a Data Set	118

Updating Multiple Data Sets	119
Validating Data	119
Zapping a CSECT	120
Chapter 9: FIXPDS Subcommand	121
<i>Active Data Sets</i>	121
Changing the FORMAT1 DSCB	123
FIXPDS of an Alternate Data Set	129
Restoring a Deleted Data Set	133
FIXPDS to Delete or Rename a Data Set	139
Appendix A: IBM Batch Utilities Comparison	143
Appendix B: ISPF Utilities Comparison	145
Glossary	147
Index	151

ABOUT THIS BOOK

This document describes Serena™ StarTool® FDM, Version 7.4, a product of Serena™ Software, Inc. The product may also be referred to as File and Data Manager or FDM.

AUDIENCE

The primary audience for this guide is installers and the Database Administrator. To install Serena™ StarTool® FDM, you should be familiar with MVS fundamentals and have experience modifying JCL.

OBJECTIVES

This manual contains information about to use StarTool FDM in a systems programming environment. To use this manual effectively, first read it as an introduction to StarTool FDM; then, refer to individual sections (especially *Chapter 8, "Examples," on page 81*) as needed in your everyday work.

MANUAL ORGANIZATION

This Chapter...	Explains...
<i>“Introduction”</i>	Contains the software, hardware and storage requirements for installing Product. It discusses security considerations and describes the contents of the distribution tape.
<i>“Product Overview”</i>	Provides a general introduction to the integrated facilities, features and functions of the product for data set and member management of IBM data sets.
<i>“Getting Started”</i>	Introduces the StarTool FDM MEMLIST facility for EDITing, BROWSing, RENAMing and DELETing commands in addition to FINDing and REPLACing character strings across selected groups of members.
<i>“Menu System”</i>	Describes the menu (or M) system, a multi-level menuing arrangement. With this system, you can type over data entry (or entry assist) panels and format and enter StarTool FDM subcommands or invoke StarTool FDM functions.
<i>“Dialog Processing”</i>	Explains how to invoke StarTool FDM from the primary options panel, from ISPF, while in READY mode, from another TSO command, and in background mode.
<i>“Batch Processing”</i>	Describes the ease of using StarTool FDM as a batch operation.
<i>“Concepts and Facilities”</i>	Offers explanations about how StarTool FDM manages data sets and its indirect method of managing members. It describes the use of the function commands, commands and subcommands, and dynamic and ISPF commands.
<i>“Examples”</i>	Provides alternatives and options to performing standard StarTool FDM tasks and the relationship between subcommands.

This Chapter...	Explains...
<i>“FIXPDS Subcommand”</i>	Explains another method for modifying the current data set while it's attributes are being changed when the data set is closed. In StarTool FDM, this is known as an alternate data set.
<i>“IBM Batch Utilities Comparison”</i>	Provide a table of IBM batch utilities and StarTool FDM batch utilities for comparison.
<i>“ISPF Utilities Comparison”</i>	StarTool FDM offers functions equivalent to most ISPF functions through its menu system. The table offers a comparison of ISPF utilities versus StarTool FDM utilities.

RELATED PUBLICATIONS

Title	Description
Serena™ StarTool® FDM User Guide	Explains StarTool FDM concepts and facilities, with emphasis on those features most useful to application programmers.
Serena™ StarTool® FDM Reference Guide	An alphabetic guide to the functions and commands of StarTool FDM.
Serena™ StarTool® FDM Messages	Explains online and batch messages for StarTool FDM, and how to deal with error situations.
Serena™ StarTool® FDM Installation Guide	Explains procedures for setting up StarTool FDM, including installation tailoring, executing the program, and installation verification.
Serena™ StarTool® FDM IMS Option	Describes the features of the IMS Option for editing IMS databases in a DLI batch environment or an online BMP environment.
Serena™ StarTool® FDM DB2 Option	Explains features of DB2 including extract and load, SQL, granting and revoking privileges, and creating, changing and dropping tables, and viewing system tables.
Serena™ StarTool® FDM System Services	Contains information about using StarTool FDM in a systems programming environment.
Serena™ StarTool® FDM StarBat Option	Describes the batch program option used for repetitive bulk data processing tasks.

Title	Description
Serena™ StarTool® FDM Extended Compare Option	Explains the functionality of Serena™ Comparex® for simple compares while using StarTool FDM.
SER10TY Installation Guide	Describes how to install and implement SER10TY licensing software for applying key CERTificates.

The Serena EPIC CD-ROM contains the Serena™ StarTool® FDM System Services Guide in Adobe® Acrobat® format. The files are password-protected (case sensitive). The CD-ROM also contains the Adobe Acrobat Reader for viewing, searching and printing the manual.

SUPPORT

Serena Software provides technical support on the Internet through the Serena eSupport self-service Web site. To access the eSupport site, go to <http://support.serena.com> and login with your ID and password to see the eSupport Customer Portal. From there, you can:

- Report new issues.
- Search our problem-tracking system for information about existing problems.
- View a knowledge base of frequently asked questions and helpful product hints.
- Query the call tracking database to obtain the current status of an open issue.
- Access our FTP server to download product fixes and documentation in PDF format.
- Subscribe to one of our mailing list servers (LISTSERVE) to receive the latest product information by e-mail.

If you need a password or ID, please e-mail or call the appropriate support department for your country.

	United States, Canada	United Kingdom, Germany, Austria, France, Switzerland, BENELUX
Support Hours	Monday through Friday 5:00 A.M. to 5:00 P.M. (Pacific Time)	Monday through Friday 7:30 A.M. to 6:30 P.M. (United Kingdom Time)
E-mail	support@serena.com	support@serena.com
URL	http://support.serena.com	http://support.serena.com
Phone	877.696.1850	44 1494 765888
Facsimile	650.522.6698	44 1494 766888
For support in all other countries, please contact your local distributor.		

INTRODUCTION



Serena™ StarTool® FDM is a multi-purpose utility program for data processing professionals, particularly systems and application programmers. It offers a new, more efficient environment in which to perform everyday tasks, while allowing you to have your previous environment available as well.

StarTool FDM runs as a normal ISPF dialog manager. It also operates as a TSO command processor in the background under the terminal monitor program. Except for the functions that are available in dialog mode only, if you are familiar with StarTool FDM operating under the ISPF dialog manager you can enter the same subcommands in TSO native mode, in a batch input stream, or under started task control.

ISPF dialog facilities are also available in batch executions of StarTool FDM if an ISPF environment is available. Line commands are not supported (except with the global APPLY command), and StarTool FDM cannot prompt for input data. You can, however, establish all of the major ISPMODE functions. A batch StarTool FDM application can save MEMLIST, LISTC/LISTF, or WORKPAD tables for later interactive use, or process multiple data sets in a table with global commands.

DATA SET PROCESSING

StarTool FDM processes disk files with partitioned, partitioned extended, sequential, VSAM, or direct organization. Any record format is acceptable. StarTool FDM assumes that any partitioned data set with record format U is a load library and other libraries are source libraries.

Members in record format U PDSE libraries (program objects) are supported in functions or subcommands that examine the internal structure of load members. Subcommands, however, that update program objects (such as REPLACE and ZAP) are not yet supported.

NEW FEATURES

Serena™ StarTool® FDM Version 7.4 contains the following new features:

Entering IMS/DLI Key Data to Retrieve Data

You can now specify the search condition (LT, LE, EQ, GE, GE) for a DBD key field while entering its search argument in any combination of subfields that may be defined in the DBD for the key field. See the StarTool FDM *IMS Option Getting Started Guide*.

DB2 Confirmation Panel

StarTool FDM now provides a confirmation panel to ensure that you want to update the DB2 table when you copy rows from one table into another.

Defining IMS Start Parameters

A new user exit for installation to specify all but the first three IMS parameters that FDM passes to IMS when it attaches DFSRRC00 to search a DLI database. See the StarTool FDM *IMS Option Getting Started Guide*.

Previous Releases Notes

You can access information on all StarTool FDM releases from the StarTool FDM main menu by choosing the NEW option.

SOFTWARE REQUIREMENTS

StarTool FDM runs under the major IBM operating systems:

- MVS/ESA (*any release*)
- OS/390 (*any release*)
- z/OS (*version 1.1*)

In addition, the following environment should be available:

- ISPF and ISPF/PDF (*version 4.1 or above*)
- TSO/E (*any release or any version*)

PRODUCT OVERVIEW

2

StarTool FDM is a comprehensive workbench for programmers, with about 100 integrated facilities that perform data set and member management for IBM data sets. StarTool FDM offers powerful facilities to manipulate data sets, individual members and groups of members using the IBM strategic ISPF dialog manager, native TSO (or READY mode), batch processing and the MVS console through a started task.

In the familiar ISPF environment, programmers have access to the latest technology. StarTool FDM implements pull-down menus, pop-up informational windows, scrollable panels, entry assist panels and field-level help to make its facilities easier to use.

StarTool FDM gives programmers an edge in managing data sets by providing facilities to logically group data sets for easy access and provides global edit and search facilities for operations on individual data sets or groups of data sets. You can perform file maintenance activities (which are usually lengthy and error prone), quickly by StarTool FDM. With data set integrity checking of StarTool FDM, trouble-shooting tools and performance enhancements, programmers gain an extra edge in meeting the technical demands in the data center.

StarTool FDM must always have a current data set; use the CHANGE subcommand to specify a different current data set. StarTool FDM supports multiple active data sets in parallel sessions. Activate and select sessions with the GO function, and terminate them with the DROP command.

ISPF programmers need no longer execute data set and member utilities to perform daily maintenance activities. StarTool FDM eliminates this work by organizing the desktop with multiple utilities. You can build data set lists, an application similar to ISPF option 3.4, using many high-level qualifiers. You can save and recall them across ISPF sessions with the StarTool FDM LISTC/LISTF application. Many batch utilities like AMASPZAP, AMBLIST, IEBCOPY, IEBCGENER, IDCAMS, IEHLIST, and IEHPROGM are part of StarTool FDM. StarTool FDM displays the linkage-edit date of load modules within a member list, maps the structure of load modules and lists the compile options for COBOL programs.

For a comparison of StarTool FDM and the IBM batch utilities, see [Appendix A, "IBM Batch Utilities Comparison," on page 143](#); to compare StarTool FDM and different ISPF utilities, see [Appendix B, "ISPF Utilities Comparison," on page 145](#).

STAR TOOL FDM BACKGROUND

StarTool FDM is a multi-purpose product that runs under MVS, designed to provide you with ample equipment to do your work with less effort.

StarTool FDM puts you in an environment. While there, you have access to new tools. StarTool FDM also has utility functions that do familiar tasks, but with less effort. You need not leave the StarTool FDM environment, and no function or tool is lost that was possessed before.

You can set up your ISPF session so that StarTool FDM supervised activities are intermixed with any other ISPF activities. Therefore, you experience no loss in either arena. The two environments blend into one. Each time there is an enhancement to StarTool FDM, there is an enhancement to your equipment.

Invoke StarTool FDM utility functions through tsubcommands. There are over one hundred subcommands in StarTool FDM, each invoking a general area of utility. Some subcommands possess more options, making more than one thousand

The installation process can restrict StarTool FDM subcommands to different classes of users on a tailored authority scale, or restrictions can be applied to selected subcommands according to a user's access to data sets. There is a default configuration provided with the product.

Debugging, linking, zapping, and module-fixing tasks become routine and error-free for programmers.

GETTING STARTED

3

StarTool FDM has a repository of software tools and facilities. Begin by using a few of the tools available, and gradually learn to use more.

For new users of StarTool FDM, become familiar with the StarTool FDM member list or MEMLIST facility. MEMLIST has a similar appearance to an ISPF/PDF member list, however, the StarTool FDM member list allows options such as ISPF EDIT, BROWSE, RENAME, DELETE, SUBMIT, and adding ISPF stats. By entering O (for OPTIONS) or M (for Menu) next to a member name, a list of available options displays.

Use MEMLIST to EDIT, BROWSE, RENAME and DELETE. You can also use StarTool FDM to display PDS directory entries in hexadecimal (DIRENTRY subcommand), or FIND and REPLACE character strings across selected groups of members. You can ZAP load modules by typing over a display of their contents, either in hex or in character. Learn about the internal makeup of your data set through the StarTool FDM USAGE and VERIFY subcommands. You can disassemble (DISASM subcommand) load modules into assembler source code, and afterwards reassemble and relink them so they are the same as they were before.

StarTool FDM lets you choose which members you want, or don't want, included in a member group. For example, you can choose most recently updated members, members that contain a specific character string, or load module members that have been zapped. Once you obtain a member group, you can add or delete members individually. After you are satisfied with your choice of group members, you can execute one operation to all members at the same time.

ENVIRONMENTS

StarTool FDM operates in four environments:

- as an interactive dialog under ISPF
- as a TSO command processor in READY mode
- as a batch dialog under the TSO Terminal Monitor Program
- as an MVS started task.

Chapter 3: Getting Started

Each mode has various advantages, but ISPF mode processing (or ISPMODE) offers the greatest programmer productivity advantages.

As an interactive dialog under ISPF, StarTool FDM combines its powerful data set and member manipulation features with the usability of ISPF. StarTool FDM takes full advantage of the facilities available in ISPF 3.3 and above. StarTool FDM is a CUA-compliant application, with pull-down menus, pop-up informational windows, scrollable panels and field-level help features designed to improve its usability. Multi-level menuing with the ability of typing over data entry panels, comprehensive tutorials and command syntax-assist panels provide the interfaces to learn the power available in StarTool FDM.

ISPMODE EXECUTION

StarTool FDM offers several selectable interface levels: standard, advanced and power. Each level or mode corresponds to a style of program use and understanding. Advanced users sometimes use facilities of StarTool FDM designed for standard users; power users exploit all aspects of StarTool FDM.

StarTool FDM provides intuitive, easy-to-use panels for standard users but some time must be spent traversing these panels and in panel prompts; StarTool FDM is less intuitive for advanced users who select assistance panels as needed but they gain in overall effectiveness by reducing prompting and increasing parallel function usage. Power users can use StarTool FDM in the most direct, effective fashion by entering subcommands directly and by using StarTool FDM in all environments.

Standard Users

Standard mode programmers use the menu system extensively. These users expect hierarchical behavior due to their ISPF background and they do not mind multiple screen prompts to establish functions. Use the SETALL command to change your interface level, but it defaults to STANDARD.

The menu system initiates member lists (from menu options 1 or 2) and data set lists (from menu option 3.4). Use line commands in functions to manipulate individual members and data sets.

Standard mode users generally understand the concepts of member groups and how to select members by contents or attributes to perform another action on them.

Some programmers choose to enter StarTool FDM to perform a specific task and exit back to their starting point. Others use StarTool FDM for almost all of their work.

When invoking StarTool FDM for a data set in ISPMODE, the first screen that displays is the Primary Options panel.

```
----- StarTool FDM Version 7.4.0 Primary Options -----  
OPTION ==>  
  
Current data set ==> LIB.CNTL  
  
  0 Parameters - Specify StarTool parameters          USERID - WSER07  
  1 Browse    - Display source data or output listings PREFIX - WSER07  
  2 Edit      - Create or change source data          TIME    - 12:26  
  3 Utilities - Perform utility functions             DATE    - 2002/04/02  
  4 Pedit     - SuperEdit Option                     JUL DATE - 2002.092  
  5 Batchjcl  - Create a background StarTool job      TERMINAL - 3278  
  A Advanced  - Menu of advanced applications        LOG PROC - ISPFD001  
  S Status    - Function status and selection menu    MVSID   - D001  
  DB DB2      - Menu of DB2 services                  APPLID  - ISR  
  IM IMS      - Menu of IMS services                  TRIAL   - 273 days  
  I# ISPF     - Stack an ISPF session (like I3.4)  
  IN Index    - Display StarTool tutorial index  
  N New       - Summarize changes for this release  
  X Exit      - Terminate StarTool
```

Chapter 3: Getting Started

From the Primary Options panel, choose option A Advanced to perform Search and Replace Functions. Then chose option 10, REPL, from the menu on the Search and Replace Selections panel. This panel is subordinate to the menu panels. Back up in the panel hierarchy with an END command to return to the previous panel. After you make another selection, Enter.

```
----- StarTool Search and Replace Selections -----  
OPTION ==>
```

Choose one of the following for a source library:

- 1 - MODULE - search system for module name
- 2 - ADDRESS - search system for module using memory address
- 3 - SEEKDS - search group of libraries for a member
- 4 - FINDDS - search group of libraries for members containing a string
- 5 - REPLDS - update members in a group of libraries
- 6 - SEEKCON - search data set concatenation for a member
- 7 - FINDCON - search data set concatenation for members containing a string
- 8 - REPLCON - update members in a data set concatenation
- 9 - FIND - search current source library for members containing a string
- 10 - REPL - update current source library members which contain a string
- 11 - IF - search current source library for members with given attributes
- 13 - VERIFY - search current source library for members with errors
- 14 - IMPACT - reconcile load library members with associated source libraries
- 15 - COMPCHK - Check a load library for compiler usage
- 16 - CSECTCHK - Check a library for like CSECTS with different compile dates
- 17 - LANGTYPE - Check a source library for languages used and mark a MEMLIST

If you select option 9, Search Current Source Library for members containing a string, and press Enter, an entry assist panel like the following sample displays. Use the entry assist panel to specify subcommand or function parameters. This formats a StarTool FDM subcommand that performs the requested function. Entry assist panels have some default information preinitialized in the panel. You can change any input field by typing over existing information. If this panel displays later in this same StarTool FDM session, previously changed values are remembered.

```

----- FIND: Search Source Members -----
OPTION ==>

- DSN=WSER07.LIB.CNTL,VOL=SER=SER005 MEM=ABC* -----
Member or member group ==> copy/ (*, start*, start:end or part/)
Find string ==> dsn= (may be 'quoted')
Format for listing ==> NUM (NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, DUMP)
Find conditions ==> (WORD, PREFIX or SUFFIX)
List all data after find ==> NO (YES or NO)
Maximum records input ==> (1 to 99999999)
Maximum records output ==> (1 to 99999999)
Maximum records to find ==> (1 to 99999999)
Records to skip ==> (0 to 99999999)
Maximum record length ==> (1 to 32767)
Columns of data to skip ==> (1 to 32767)
    
```

On color terminals, input fields (just after each ==> marker) display in red by default. To position the cursor to any field on an entry assist panel, use the tab down key or the arrow keys. You can blank out most fields in an entry assist panel with the erase-EOF key and if there is a choice of keywords, you need only type over enough to make your choice unambiguous. For example, if LBLOCK displays and your choices include NUM, SNUM and NONUM, changing the entry to noLOCK is sufficient to specify NONUM.

For information while in an entry assistance panel, press HELP for tutorials on that subcommand. Some input arrow (==>) fields on entry assist panels (like the “Member or member group” line above) are marked in a special color (green by default) to indicate that field-level help is available for that field. Get information by placing the cursor in the field and pressing the HELP key (PF1). If the cursor is in another field that does not have field-level help, the HELP command provides normal subcommand tutorials.

After you press ENTER on an entry assist panel, the system goes to the log where the formatted subcommand displays prefixed by >-----> (this demonstrates StarTool FDM subcommand syntax), followed by the output of the subcommand.

Chapter 3: Getting Started

Examine the output of the subcommand in the log, then press END to return to the entry assist panel and make corrections and retry the subcommand. Each time you execute the END command, the system backs up one level in the panel hierarchy. If you want to go to the menu primary panel directly, type the MENU command or type a chained command such as M.3.4 or its equivalent, 3.4 (numbered commands are references to the menu system when operating in the log or any other table panel).

```
FUNCTIONS CONTROL DSN CMDS MEM CMDS A-M MEM CMDS N-Z DEFAULTS FEATURES
-----
----- ISPMODE Session# 1 Log# 1 -- ROW 162 TO 179 OF 245
COMMAND ==> SCROLL ==> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006 MEM=COPY/ -----
>----->Find copy/ 'dsn=' NUM CAPS

** FIND IEBCOPY
000900 //SYSUT1 DD DSN=PDSE.MAPS,DISP=OLD,UNIT=3480,
001100 //SYSUT2 DD DSN=SER09.MODS2.ASM,DISP=SHR
PDS142I 10 lines in this member
PDS146I 2 strings found

** FIND IEBCOPYA
000270 //SYSUT1 DD DSN=QUICKREF,DISP=OLD,UNIT=3480,
000290 //SYSUT2 DD DSN=SER07.QUICKREF.JCL,DISP=(,CATLG),
PDS142I 14 lines in this member
PDS146I 2 strings found

** FIND IEBCOPYB
000700 //SYSUT1 DD DSN=OXTS207PDSTOOLS.STOW.COPYDAT.TAPECPY,UNIT=3480,
000800 //SYSUT2 DD DSN=SER07.LIB.STOW,DISP=(,CATLG,DELETE),UNIT=SERDA,
PDS142I 14 lines in this member
```

In the figure above, the data from “>----->Find...” through the last line in the display is log data. The log tracks subcommands entered and the output from subcommands in a scrollable ISPF table. An ISPF command such as HELP is not tracked in the log, but a StarTool FDM subcommand such as FIND and the results of the FIND are in the log.

Subcommands are identified in the log by a >-----> string. Even though the log supports up to 159 characters in each log line, if an echoed subcommand does not fit within the first 79 characters of a log line, continuation lines are added as needed. Each continuation line is identified with an initial - character.

Several facilities are available for manipulating the log.

- You can manipulate the output from the last subcommand in an edit session with the EDITLOG command.

- Use the OUTPUT command to direct the log to a data set or a system printer .
- Use the EDITTBL command to manipulate the entire log in an edit session.
- Search any character string in the log with the F command. Then press RFIND for repeat searches.
- Select other views of the log by pressing LEFT or RIGHT.
- If the cursor is positioned over a StarTool FDM message (PDSnnn) in the log, use the RCHANGE command for an explanation.

Advanced Users

More advanced programmers use menu system facilities selectively. For example, an advanced user enters M.ML to get an entry assist panel to select members with certain attributes for a member list. Advanced level is similar to standard level except that more detailed subcommand syntax is available and the fewer assumptions are made in the panel interface. Use the SETALL command to set your "interface level" to ADVANCED.

Advanced users also selectively use the syntax assist panels (for example, O.ML), the CUA action bars and the help facilities to for help for a CUA action bar item, a pull-down CUA menu item or a table row item.

Advanced users use the major ISPMODE functions heavily (MEMLIST, CMDTBL, CSECTS, ZAP, LISTA, DDNAME, LISTC, LISTF, LISTV, PBROWSE, PEDIT and WORKPAD) and move between these parallel functions. They save and recall MEMLIST, LISTC/LISTF and WORKPAD tables and initiate global commands against data set lists in a LISTA/DDNAME, LISTC/LISTF or WORKPAD table.

Chapter 3: Getting Started

Advanced users, like standard users, enter StarTool FDM in the primary selection panel. If you go from the menu panel (with option 8 or the LOG command), the following panel displays.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-----
----- ISPMODE Session# 1 Log# 1 -----  ROW 1 TO 15 OF 15
COMMAND ==> _                               SCROLL ==> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006 MEM= -----
PDS100I StarTool/SuperEdit -- Version 7.4.0  2000.001

Proprietary software product of SERENA Software Intl.
Phone (650)696-1800 OR FAX (650)696-1776
LICENSED TO: your corporate name/trial offer expires ...
             your city, state, zip/agent to contact for license ...
All other rights reserved - use of this software
product by unauthorized persons is prohibited.
PDS200I DISP UNIT      RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390      FB           80  13680   1X   100          2    50 TRK      17

PDS572W This data set has 2 free tracks

PDS300A ENTER OPTION -- DSN=SER07.LIB.CNTL,VOL=SER=SER006 MEM=
***** BOTTOM OF DATA *****
```

This panel is divided into several areas. The first line or Action Bar is the CUA pull-down system that displays available commands when activated. Use the tab key or the arrow keys to place the cursor over an action bar topic and activate it by pressing the ENTER key (you can turn off CUA action bars for StarTool FDM processing for the remainder of a session or for all future sessions with the SETALL command).

The FUNCTIONS menu was selected in the preceding panel displaying the following panel.

```

FUNCTIONS CONTROL          DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
+-----+-----+-----+-----+-----+-----+
| 1. CAX                   |          |          |          |          |          |
| 2. CSECT                 | L,VOL=SER=SER006 MEM= |          |          |          |
| 3. GO                     | Version 7.4.0 2000.001 |          |          |          |
| 4. HEX                   |          |          |          |          |          |
| 5. LISTA                 | E PRODUCT OF SERENA SOFTWARE |          |          |          |
| 6. LISTC                 | OR FAX (650)696-1776 |          |          |          |
| 7. LISTFILE              | orporate name/trial offer expires ... |          |          |          |
| 8. LISTV                 | ity, state, zip/agent to contact ... |          |          |          |
| 9. MASK                  | SERVED - USE OF THIS SOFTWARE |          |          |          |
| 10. MEMLIST              | IZED PERSONS IS PROHIBITED. |          |          |          |
| 11. NUCMAP               |          |          |          |          |          |
| 12. STATUS               | RECFM LRECL BLKSIZE  ALLOC TRK FREETRK SECONDARY FREEDIR |          |          |          |
| 13. WORKPAD              | FB          80 13680  1X 100          2 50 TRK          17 |          |          |          |
| 14. LOG                  |          |          |          |          |          |
| 15. ZAP                  | et has 2 free tracks |          |          |          |
+-----+-----+-----+-----+-----+
PDS300A ENTER OPTION -- DSN=SER07.LIB.CNTL,VOL=SER=SER006 MEM=
***** BOTTOM OF DATA *****

```

The cursor is on the underline symbol next to item 1. CAX. To exit from this menu use the END key. To get brief help information on a menu item's function, place the cursor next to an item in the CUA menu box and press HELP. A pop-up window appears with a short explanation. Exit from the pop-up window with the END key.

To activate a menu item, type the item number without moving the cursor or move the cursor to the option and press ENTER. After selecting an item, its syntax assist panel appears. Syntax assist panels display all optional suboperands pertinent to a subcommand or function. If you want additional information, press HELP while the syntax assist panel displays to go to the StarTool FDM tutorials. Exit from help tutorials with the END key.

The command line operates like the standard ISPF command line. Use it to enter primary commands (subcommands and functions) pertaining to the entire data set, such as searching a group of members, creating and maintaining data set lists or controlling the environment in which StarTool FDM operates. Type INDEX as a primary command to display the tutorial index of all available subcommands and functions.

Chapter 3: Getting Started

In the preceding Sample StarTool FDM Log panel, the data from the PDS1001 message through the line reading `*** BOTTOM OF DATA ***` is log data. The log tracks subcommands entered and the output from subcommands in a scrollable ISPF table.

Select data from the log by placing the cursor over the data and pressing the RCHANGE key.

- For a subcommand for optional modification and reuse.
- For a data set name select it for optional modification and activation.
- For a StarTool FDM message (PDS`nnn`), a full explanation of that message is in the log.

Power Users

Power mode programmers fully exploit all aspects of StarTool FDM. Many of these users spend more than 90% of their working time in an active StarTool FDM session with a correspondingly large boost in their programming effectiveness. Power level is similar to advanced level except that power users bypass the primary menu panel. Use the SETALL command to set your "interface level" to POWER.

Power users establish multiple GO sessions and move between parallel data set sessions as needed. Some power users also split their screens to have another application or a different StarTool FDM session on the other side.

Power users generally know StarTool FDM subcommands and functions well enough so that they rarely need assistance. Power users know how to enter a chained command (for example, M.ML or O.ML) to display an entry assist or syntax assist panel for a subcommand. These users also program their own dynamic primary commands and dynamic line commands to launch other applications from StarTool FDM.

Power users are able to use StarTool FDM in ISPMODE, line mode, batch mode and STC mode. When operating in ISPMODE, they minimize prompting and "screen flopping" and maximize parallel activity and the data available in each table panel.

LINE MODE

Much of StarTool FDM refers to ISPMODE operation. A large portion of StarTool FDM is available in line mode.

StarTool FDM operates outside of ISPF in line mode under the READY prompt of TSO. As a TSO command processor, StarTool FDM supports all subcommands that do not require ISPMODE facilities. TSO CLISTS and REXX command lists are built to establish a StarTool FDM environment and process subcommand scripts.

Line mode of StarTool FDM does not involve ISPF. It is the direct operation of StarTool FDM under TSO, and all responses from StarTool FDM are written to the terminal with the PUTLINE interface of TSO. Because the PUTLINE terminal interface is used, StarTool FDM output is directed to SYSOUT or to an external data set. (See *"CONTROL DSN" option in the Reference section of StarTool FDM Reference Guide*. This command allows you to specify an external data set that contains the output of StarTool FDM subcommands from your session.)

If ISPF is available, you can toggle between line mode and ISPMODE operations. Under ISPMODE, the SUSPEND command immediately places you into direct TSO command mode operation (this is, line mode). If you enter StarTool FDM directly under ISPF with the command:

```
TSO  STARTOOL  'data.set.name'  XISPMODE
```

this forces initial entry to StarTool FDM in line mode. Then, the ISPMODE subcommand places you immediately into ISPMODE, provided ISPF is available and the panel and message data sets are allocated.

The value of StarTool FDM in line mode is that you can obtain most of the StarTool FDM utility function, even without ISPF. There are situations when full-screen operation is unavailable such as when running in batch mode, or when executing TSO commands from an operator console. The PDSE started task is a subsystem under MVS that permits the execution of TSO commands in line mode from an operator console (without JES2, VTAM or TSO). You can run StarTool FDM under the PDSE started task such as for recovery purposes when the JES has to be fixed, for utility operations under an operator console or for more general automation tasks.

Another useful facility available under line mode is IND\$FILE. To transmit data to or from a personal computer, type SUSPEND to go to line mode, switch to a DOS or OS/2 session, type SEND or RECEIVE commands, switch back to your TSO session, and type ISPMODE to resume ISPMODE processing.

Chapter 3: Getting Started

The OPTIONS subcommand lists all StarTool FDM subcommands. Type subcommands on successive lines, or stack them on a single line using the "field mark" character as a separator. All StarTool FDM subcommands except those explicitly having to do with ISPMODE are available.

In line mode, with the aid of the TSO subcommand of StarTool FDM, you can mix subcommands and ordinary TSO commands to perform complex operations. For example, suppose you want to move all non-reentrant members of a load library to one data set, and reentrant members to another data set, and then rename the original data set. Type:

```
STARTOOL  'first.dataset'  XISPMODE  Invokes StarTool FDM in line mode
                                         pointing to data set
                                         'first.dataset'.

IF : RENT THEN(SUBLIST)  Starting from all members, create a subgroup of
                           members consisting only of the reentrant
                           modules.

COPY * 'second.dataset' NEW VOL(volser)  Creates a new data set
                                         'second.dataset' on volume volser
                                         having the same space and
                                         attributes as 'first.dataset'.
                                         Invoke IEBCOPY or its equivalent
                                         to copy only the members in the
                                         current subgroup - the reentrant
                                         ones - from 'first.dataset' to
                                         'second.dataset').

IF : NORENT THEN(SUBLIST) Starting from all members, create a subgroup of
                           members consisting only of the non-reentrant
                           modules).

COPY * 'third.dataset' NEW VOL(volser)  Creates a new data set
                                         'third.dataset' on volume
                                         volser having the same space and
                                         attributes as 'first.dataset'.
                                         Then invoke IEBCOPY or its
                                         equivalent to copy only the
                                         members in the current subgroup -
                                         the non-reentrant ones - from
                                         'first.dataset' to
                                         'third.dataset').

CHANGE 'second.dataset'  Changes the StarTool FDM current data set to
                           to 'second.dataset').

AT : SHORT              Display attributes of all members to make sure
                           they are reentrant).

CHANGE 'third.dataset'  Changes the StarTool FDM current data set to
                           'third.dataset').

AT : SHORT              Display attributes of all members to make sure
                           they are non-reentrant).

TSO RENAME 'first.dataset' 'xxx.first.dataset'  Invokes the TSO RENAME
                                                   command to rename and
```

```
recatalog
'first.dataset').
END Exits from StarTool FDM).
```

In another example, use StarTool FDM to replace one string with another string over selected members. For example, suppose you replace data set prefix SYS2 with SYS3 in all occurrences in SYS1.PROCLIB except for the members beginning with CICS, as follows:

```
STARTOOL 'SYS1.PROCLIB' XISPMODE Invokes StarTool FDM in line mode.
SUBLIST : EXCLUDE (CICS*) Make a group consisting of all members
except for members beginning with CICS).
REPLACE * /SYS2./SYS3./ WRITE Replaces the data set prefix SYS2. with
SYS3. only in members of the current
subgroup. The keyword WRITE makes the
changes final. Otherwise, get a dry run).
END Exit StarTool FDM execution.
```

You can also perform these examples in ISPMODE. Many utility functions of StarTool FDM are available in line mode. These are useful when full screen facilities are not available.

BATCH MODE

StarTool FDM operates under the batch Terminal Monitor Program (TMP). Execution is very similar to TSO command processor mode except that user interaction is not supported. To execute a script in batch, place several subcommands in the input stream or in CLIST or REXX command lists. This mode of execution is very effective for performing regular maintenance functions such as compressing libraries, transferring test members to production libraries, validating libraries and searching for unusual members or data sets.

ISPMODE functions are available in batch mode; however, panel prompting is not supported and line commands are supported only by the global APPLY command. Multiple data sets are easily processed using ISPMODE facilities in batch through global commands.

For more information on batch execution, see [Chapter 6, "Batch Processing," on page 65](#).

STARTED TASK MODE

StarTool FDM operates as an MVS started task called PDSE (subsystem names such as PDSE and JES2 are limited to four characters). Line mode subcommands are issued through an MVS console (or SDSF) and are processed by the PDSE subsystem. This type of execution adds recovery capabilities for system outages even in those instances when JES, VTAM or TSO are unavailable.

For more information on using StarTool FDM in started task mode, see the PDSE Started Task chapter in the Serena™ StarTool® FDM *Installation Guide*.

INVOKING STAR TOOL FDM

Invoke StarTool FDM as a TSO Command Processor or as an ISPF dialog. If ISPF data sets are allocated containing the StarTool FDM panel and message members, you can get control in either mode, and you can toggle between the two modes.

StarTool FDM must always point to a data set. This data set is referred to as the current data set of the StarTool FDM session. StarTool FDM supports multiple current data sets in ISPMODE through its GO function, but there must always be at least one current data set to which StarTool FDM points. You can change the current data set with the CHANGE subcommand. For StarTool FDM to gain control under ISPF, enter:

```
TSO  STARTOOL  'data.set.name'
```

If you omit the data set name, StarTool FDM uses the last data set from the previous StarTool FDM session.

Normally, you invoke StarTool FDM as an ISPF dialog in ISPMODE. You can also invoke StarTool FDM directly in ISPMODE through a primary panel option. The introductory dialog panel appears. To switch operation to line mode, use the SUSPEND command. You can reenter ISPMODE with the ISPMODE subcommand allowing you to toggle between the two modes of operation.

ISPMODE operation of StarTool FDM consists of a sequence of panel displays that you invoke one from another, depending on choices presented, and from which you decide where to navigate. StarTool FDM ISPMODE dialog is no different from any other ISPF dialog. The normal rules and functions of ISPF apply.

The introductory panel asks you to pick a data set that becomes the current data set for the StarTool FDM session. After you select a current data set on the introductory panel and press Enter, the menu primary panel displays if the standard interface is in use. Select option 8 to display the log panel of ISPMODE.

You can also invoke StarTool FDM from an ISPF data list (ISPF option 3.4) by typing STARTOOL or PDSE as a line command for a data set. StarTool FDM honors the data set volume name. (Normally, ISPF line commands do not honor the volume name.) In general, ISPF and StarTool FDM help each other and do not interfere with one another.

Besides the log, StarTool FDM offers other table displays and any of them can be active at the same time in parallel mode. When you type the STATUS primary command, a panel appears showing the status of all ISPMODE table functions. From this status panel, you can immediately pass to any of the table displays by typing the appropriate command.

Single Subcommand Mode

You can invoke StarTool FDM in single subcommand mode by including a subcommand and any required operands after the data set name. StarTool FDM only performs the single subcommand and terminates.

ISPMODE subcommands such as MEMLIST, ISPMODE and ISPXEQ are exceptions. They invoke StarTool FDM in an ISPF dialog mode, as expected. Also, XISPMODE is supported as a mode command so that you can invoke StarTool FDM in line mode from an ISPF environment. Following is an example of invoking StarTool FDM in single subcommand mode from an ISPF environment (from READY mode, omit TSO).

```
TSO  STARTOOL  data.set.name copy  :  new.data.set.name  new
```

Note This is a special subcommand mode for performing only a single subcommand. ISPF services are not available and StarTool FDM operates in line mode only. Also, no YES/NO prompting is provided; instead, YES responses are assumed.

When you invoke StarTool FDM in single subcommand mode, the return code is set to the numeric value of the first warning or error message encountered. For example, if the subcommand encountered a PDS820E message, the return code is set to 820 in decimal.

License Information

When you invoke StarTool FDM, license information displays, as shown below to identify the program, release number, and release date.

```
PDS100I StarTool/SuperEdit -- Version 7.4.0 2000.001

Proprietary software product of SERENA Software Intl.
Phone (650) 696-1850 OR FAX (650) 522-6698
LICENSED TO: your corporate name/trial offer expires ...
             your city, state, zip/agent to contact for license ...
All other rights reserved - use of this software
product by unauthorized persons is prohibited.
```

MENU SYSTEM

4

The MENU system provides multi-level menuing with the ability to type over data entry (called entry assist panels). Using these panels, you can format and enter StarTool FDM subcommands or invoke StarTool FDM functions.

The menu system is similar to ISPF.

PRIMARY OPTIONS MENU

The StarTool FDM Primary Options panel displays on entry to StarTool FDM:

```
----- StarTool FDM Version 7.4.0 Primary Options -----
OPTION ==>

Current data set ==> LIB.CNTL

 0 Parameters - Specify StarTool parameters          USERID - WSER298
 1 Browse     - Display source data or output listings PREFIX - WSER298
 2 Edit       - Create or change source data         TIME    - 16:45
 3 Utilities  - Perform utility functions            DATE    - 2002/03/28
 4 Pedit     - SuperEdit Option                      JUL DATE - 2002.087
 5 Batchjcl  - Create a background StarTool job      TERMINAL - 3278
 A Advanced  - Menu of advanced applications        LOG PROC - ISPFD001
 S Status    - Function status and selection menu   MVSID   - D001
 DB DB2      - Menu of DB2 services                 APPLID  - ISR
 IM IMS      - Menu of IMS services                 TRIAL   - 3 days
 I# ISPF     - Stack an ISPF session (like I3.4)
 IN Index    - Display StarTool tutorial index
 N New       - Summarize changes for this release
 X Exit      - Terminate StarTool
```

From anywhere in the StarTool FDM menu system, you can display the primary menu panel by typing M. followed by any data for the menu system. For example, type M.FIND to go to an entry assist panel for the FIND subcommand.

Chapter 4: Menu System

From any function in StarTool FDM, use numbers to reference the menu system with any of the short-hand conventions listed below:

- 3.4** equivalent to M.3.4
- =3.4** equivalent to M.3.4 (see “Internal Jump” in SETALL)
- M.FIND** equivalent to M.12.FIND since option 12 is a subcommand reference.
- FIND** equivalent to M.12.FIND because most subcommands entered without operands implicitly invoke the menu system.

M (for MENU) is also available as a line command. From a source or load member list, type M as a line command for a member to get a directory of available line commands. When you select one of these, the entry assist panel for that subcommand displays. You can also type a chained line command such as M.FI (for FIND) to bypass the directory prompting panel.

The menu system makes the power of StarTool FDM available to all programmers. You can transfer your ISPF skill set directly into the StarTool FDM environment with no additional training. With the entry assist panels, you can type over fields to reformat panels and enter subcommands to perform StarTool FDM functions. The primary menu panel is logically divided into option groups that perform different types of functions, as explained below.

- **0 through 5** PARAMETERS, BROWSE, EDIT, UTILITIES, PEDIT and BATCH are similar to corresponding ISPF options. Results display with the EDITLOG facility. Control remains in the menu system. PEDIT is a part of SUPEREDIT, an alternative to the ISPF edit command.
- **A** ADVANCED provides a menu of advanced applications.
- **S through X** STATUS, DB2, IMS, ISPF, INDEX, NEW and EXIT provide various support services. Control returns to the primary options panel for each option except for EXIT.

Individual items from the primary options panel are:

- **OPTION ==>** a field for entering commands. Type a StarTool FDM function, command, dynamic command, subcommand or a numbered option.
- **Current data set:** a field to change to a different current data set. Tab to this field and type any one of the following:

- **data set name** formatted as for an ISPF "Other data set" field. Uncataloged data sets are not supported here but you can use option 9 for these data sets.
- **FILE(ddname)** to change to a preallocated DDNAME. Type a NUMBER(*n*) parameter to select the *n*th data set in a set of concatenated data sets.
- * to change to the previous data set.
- **blank** to request a data set prompting panel.
- **0 - Parameters** displays a menu for the StarTool FDM SET panels.
- **1 - Browse** displays a method to specify a new current data set for StarTool FDM. If you specify a member group name, StarTool FDM builds a member list of those members; otherwise, ISPF Browse is invoked on a single member.
- **2 - Edit** provides a method to specify a new current data set for StarTool FDM. If you specify a member group name, StarTool FDM builds a member list of those members; otherwise, ISPF Edit is invoked on a single member.
- **3 - Utilities** provides a method to specify a new current data set for most subordinate utility functions. StarTool FDM performs functions corresponding to the equivalent ISPF 3.x options.
- **4 - Pedit** provides an alternative to the ISPF editor. PEDIT supports VSAM and direct data sets, load members, copybook data overlays, selective record edit, multiple edit sessions, large data sets, physical record access and long records (more than 255 characters).
- **5 - Batch** creates background jobs. Controls are available to build batch JCL from allocations in the DDNAME/LISTA table, build a StarTool FDM batch job from subcommands in the log, manage the index for disk volumes, set the VTOC DIRF bit and manage the DDNAME/LISTA table.
- **S - Status** stacks an ISPF dialog. For example, I3.4 is equivalent to ISPF 3.4
- **DB – DB2** displays a menu of DB2 services.
- **IM - IMS** displays a menu of IMS services.
- **I# - ISPF** stacks an ISPF dialog. For example, I3.4 is equivalent to ISPF 3.4
- **IN - Index** enters the StarTool FDM tutorial index.
- **N - New** displays a summary of changes for the current release and older releases.
- **X - Exit** terminates StarTool FDM (displays the number of current active GO sessions).

ADVANCED OPTIONS MENU

Access the Advanced Options menu by selecting option A from the Primary Options menu:

```
----- Advanced Options -----  
OPTION ==>  
  
 6 All           - Create a MEMLIST with all members  
 7 Output        - Output LOG Table  
 8 Log           - View StarTool session log  
 9 Activate      - Activate data sets and GO sessions  
10 Search        - Search and Replace functions  
11 Recover       - Recover and Repair functions  
12 Source PDS    - Services for WSER298.LIB.CNTL,VOL=SER=SMS013 MEM=  
13 Examine       - Examine current environment and system resources  
14 Profman       - Manage saved member and data set lists  
15 Global        - Global processing of data set and volume lists  
16 Comparex      - Use the Comparex Dialog for WSER298.LIB.CNTL  
17 CBSselect     - Use copybooks to specify record selection when copying  
18 CBSselect     - Use a previous saved copybook selection table
```

Note: Comparex is a separately licensed compare product from Serena

The Advanced Options menu is divided into option groups, as follows:

- **6 through 8** All, Output and Log perform specialized StarTool FDM functions. Control normally navigates to tutorial panels first; however, you can suppress them (see SETPANEL).
- **9 through 11** Activate, Search and Recover provide easy access to StarTool FDM subcommands and functions. Control is received in the log where results display. Press END to return to the last panel displayed. Then, retype any fields desired and press Enter to reexecute a subcommand or continue backing up in the panel hierarchy by pressing END.
- **12** This option varies according to the active data set type (Direct, Sequential, Source PDS/PDSE, Load PDS/PDSE or VSAM). Only subcommands supported for the active data set type are listed for this option. Option 12 is automatically selected from the primary options panel or from any of the StarTool FDM functions if you type a subcommand without operands.
- **13 through 15** Examine, Profman and Global provide advanced StarTool FDM subcommand and function support. Control is received in the log where results display. Press END to return to the last panel displayed. Then, retype any fields desired and press Enter to reexecute a subcommand or continue backing up in the panel hierarchy by pressing END.

Following are descriptions for each option:

- **6 - All** builds a member list containing all data set member. This is equivalent to the ALL command.
- **7 - Output** outputs the current dialog table. This is equivalent to the OUTPUT command.
- **8 - Log** switches from the menu panel to the log display. This is equivalent to the LOG command.
- **9 - Activate** controls GO sessions and allows changing the current data set.
- **10 - Search** provides dialogs to search and update several different types of data.
- **11 - Recover** provides several dialogs to recover and repair the current data set.
- **12 - Subcommands** provides a menu for subcommand entry assist panels based on the current data set organization.
- **13 - Examine** controls several specialized subcommands that display environmental information.
- **14 - Profman** manages saved MEMLIST, LISTC/LISTF and WORKPAD tables.

Chapter 4: Menu System

- **15 - Global** controls global commands for DDNAME/LISTA, LIST/LISTF, LISTV and WORKPAD. You can create or extend the tables and initiate global processing.
- **16 - Comparex** invokes the Serena™ Comparex® product, which is separately licensed.
- **17 - CBSelect** allows you to copy selected records using copybook variable names.
- **18 - CBSelect** allows you to use a saved copybook selection table to copy selected records.

NAVIGATING IN STAR TOOL FDM

From StarTool FDM table panels, an END command invokes the primary options panel allowing you to choose your next option.

After you enter data on an entry assist panel, the panel logic formats a StarTool FDM subcommand or function and enters the command. This places you in one of the StarTool FDM functions so that you can review the results of the subcommand. For StarTool FDM subcommands, you go to the log where the formatted subcommand and its output display.

After you review the results of the subcommand, press END to return to the last menu panel displayed since the menu system operates in hierarchical mode (normally, StarTool FDM operates in parallel mode). Hierarchical mode is maintained until you enter a parallel mode command. For example, if you are examining the log, you can type one of the following commands without interrupting hierarchical mode: F, OUTPUT, RIGHT, LEFT, UP, DOWN, EDITLOG, EDITTBL, X. If you enter a mode switch command or function command like ML or a subcommand like FIND, you begin operating in parallel mode. This means that a following END command goes to the primary options panel instead of the last menu panel displayed.

In the menu panels, an END command backs up a single panel. If you enter END from the primary options panel, the current GO session is dropped. StarTool FDM asks that you confirm termination of each GO session if your SETALL value for Termination Prompting is YES. If you enter END for the last GO session, you are always asked to confirm termination since you are leaving the StarTool FDM environment. If you want to terminate StarTool FDM with no prompts, type an X command from the primary options panel.

Jump commands (like =3.4) are supported and behave as expected for a primary panel (jump commands request StarTool FDM options) unless Internal Jump in SETALL is set to NO (then, jump commands terminate StarTool FDM and request ISPF options). StarTool FDM also supports internal jump commands if you enter a blank between the equal sign and the number (for example, = 3.4).

- If Internal Jump is YES:
 - =X** from any StarTool FDM panel terminates StarTool FDM
 - =3.4** executes StarTool FDM option 3.4
 - RETURN** goes to the primary options panel
- If Internal Jump is NO:
 - =X** from any StarTool FDM panel terminates StarTool FDM and executes ISPF option X
 - =3.4** terminates StarTool FDM and executes ISPF option 3.4
 - RETURN** terminates StarTool FDM

OPTIONS FOR SYSTEMS PROGRAMMERS

In this chapter, only options specifically used by systems programmers are covered. For detailed information of the other options, refer to Serena™ StarTool® FDM 7.4 *User Guide*.

Option 13 - Examine

Use Option 13, Examine Current Environment, to:

- locate MVS components
- display your TSO session
- display disk space for selected volumes
- display users allocated to a specific data set
- display status of StarTool FDM functions
- examine StarTool FDM installation parameters
- display current catalogs

Chapter 4: Menu System

The following screen is an example for Option 13 - Examine Current Environment.

```
----- StarTool Examine Current Environment -----  
OPTION ==> listenv  
  
Choose one of the following:  
  
  1 - FINDMOD - Find a system module  
  2 - LISTA   - Display allocated DDNAMEs  
  3 - DDNAME  - Display allocated DDNAMEs matching a mask  
  4 - LISTV   - Display selected system volumes  
  5 - SVCMAP  - Display a system SVC  
  6 - WHOHAS  - Display users allocated to a data set  
  7 - STATUS  - Display status of ISPMODE functions  
  8 - LISTENV - List the current hardware and software environment  
  9 - DEFAULTS - List StarTool customized defaults  
 10 - RESTRICT - List StarTool restricted subcommands  
 11 - CATALOGS - List current active catalogs  
 12 - SYSTAT  - Invoke Sys/Stat from Advent Software for System Status  
 13 - VUUSE   - Display volume usage, space and statistics
```

The options on this menu are:

- **1 - FINDMOD** searches for system modules. You can examine redundant copies of members in the linklist or LPALIST.
- **2 - LISTA** displays allocated DDNAMES. You can filter the displayed DDNAMES on several different criteria.
- **3 - DDNAME** displays allocated DDNAMES filtering by a mask name.
- **4 - LISTV** displays disk volume space and attributes. You can filter the displayed volumes using a volume name mask.
- **5 - SVCMAP** displays system SVCs from the executing MVS system.
- **6 - WHOHAS** displays users allocated to a data set.
- **7 - STATUS** displays the current status of all StarTool FDM ISPMODE functions. You can switch to any other function from STATUS.
- **8 - LISTENV** displays the current software and hardware environment. Use the CONTROL subcommand with a LISTENV operand.
- **9 - DEFAULTS** displays StarTool FDM customized defaults. Use the CONTROL subcommand with a DEFAULTS operand to display options selected in the PDS#OPT4 module.
- **10 - RESTRICT** If permitted at your installation, displays the names of subcommands that are restricted for your session. Use the CONTROL subcommand with a RESTRICT operand.
- **11 - CATALOGS** displays current active system catalogs. Use the CAX function.
- **12 - SYSTAT** invokes a software product called Sys/Stat from Advent Software if available at your installation.
- **13 - VUSE** displays disk volume usage, space and statistics.

Option 13.8 - Current Hardware and Software Environment

After you press ENTER on the Examine Current Environment panel, the results of the CONTROL LISTENV subcommand display.

StarTool FDM identifies the hardware and software environment with a display of the different types of MVS system storage. Areas that are not present in a system do not display (for example, a FLPA area was not established for this example, but an E-FLPA area was established).

Chapter 4: Menu System

This is an excerpt from a log created with Option 13.8:

```
PDS100I PDSE/SuperEdit -- Version 7.4.0 2000.001

PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(8)
PDS036I Largest free storage area is 3020K
PDS046I Largest area above the line is 1891M

PDS280I System serial:05905; CPU type:5995
PDS281I Active CPUs:4 5 6 7
PDS282I SMF ID:SCU1; System mode:ESA/370
PDS283I Maintenance data:SP4.3.0 HBB4430
PDS284I IPL date:97/04/14 97.084; Time:06:11
PDS285I IPL type:CLPA; Volume:ES1RES; UNIT:54A
PDS286I Master catalog dsname:SYS1.SCU1S11.ICFMCAT; Volume:ES1CAT;
UNIT:541
PDS287I OS/390 1.2.0; DFSMS 1.3.0; DFSMSshm; DFSMSdss; DFSMSrmm; ISPF
4.3;
PDS287I VTAM 4.3; TSO/E 2.05.0; RACF 2.02.0;
PDS288I Current NUCLEUS ID:1; I/O CONFIG ID:03

PDS052I Real storage is 472M; extended storage is 512M
PDS053I LOAD parameter is '054100M '

PDS059I Storage map          START          END          SIZE
PDS059I -----          -
PDS059I E-PRIVATE          09100000    7FFFFFFF    1,948,672K
PDS059I E-CSA              042CB000    090FFFFFFF    80,084K
PDS059I E-MLPA             042CA000    042CAFFFF    4K
PDS059I E-FLPA             042C7000    042C9FFF    12K
PDS059I E-PLPA             02837000    042C6FFF    27,200K
PDS059I E-SQA              01A9F000    028361DF    13,920K
PDS059I E-NUCLEUS (R/W)    012E3000    01A9EFFF    7,920K
PDS059I E-NUCLEUS (R/O)    01000000    012E24FF    2,956K
PDS059I ----- 16 Megabyte Boundary Line -----
PDS059I NUCLEUS (R/O)      00FDD000    00FFFFFFF    140K
PDS059I NUCLEUS (R/W)      00F94000    00FDCD8F    292K
PDS059I SQA                00E84000    00F93FFF    1,088K
PDS059I PLPA               00C87000    00E83FFF    2,036K
PDS059I MLPA               00C84000    00C86FFF    12K
PDS059I CSA                00800000    00C83FFF    4,624K
PDS059I PRIVATE            00005000    007FFFFF    8,172K
PDS059I V=R AREA           00005000    00024FFF    128K
PDS059I SYSTEM             00001000    00004FFF    16K
```

PDS059I PSA

00000000 00000FFF

4K

Option 13.9 - Customized Defaults

The following is an example of the output from option 13.9 (or CUSTOMIZED DEFAULTS).

StarTool FDM:

- identifies its current level (PDS100I message)
- displays global operands (PDS030I message)
- displays input buffering (PDS031I message)
- displays largest free storage areas (PDS036I and PDS046I messages), before beginning the installation defaults with the PDS037I message.

The PDS037I message identifies the name of the defaults module (PDS#OPT4) and the date and time of its assembly. Use this information to verify that the correct version of PDS#OPT4 is being used to set StarTool FDM defaults.

Chapter 4: Menu System

StarTool FDM displays installation parameters and the associated choices. Use this list as a summary of the choices made when you installed StarTool FDM. This information is often useful in debugging environmental problems in StarTool FDM.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-----
----- ISPMODE Session# 2 Log ROW 3,149 TO 3,167 OF 3,213
COMMAND ==>                                SCROLL ==> CSR
- DSN=WSER09.PDSES.ASM,VOL=SER=STR840  MEM=(@DIACLIK  -----
>----->Control defaults

PDS100I PDSE/SuperEdit -- Version 7.4.0  2000.001

PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(8)
PDS036I Largest free storage area is 1128K
PDS046I Largest area above the line is 1915M

PDS037I Installation defaults from PDS#OPT4 97/04/14 09.49:
Access control method          NONE
Security tables                 SYSTEMSE SYSTEMSN APPLEXP OTHERS
DSN default format             MSG
COMPRESS SUMMARY/LIST         SUMMARY
COPY SUMMARY/LIST              LIST
COPY SHR/OLD allocation        SHR
COPY temporary unit            SYSDA
COPY permanent unit            SYSALLDA
CONTROL "NO" defaults          0
```

DIALOG PROCESSING

5

StarTool FDM is a TSO command processor that may not always have access to ISPF dialog services. Start StarTool FDM from the following environments:

StarTool FDM Primary Options panel

Invoke StarTool FDM as a full ISPF dialog. StarTool FDM locates or creates any required command processor control blocks.

ISPF

Invoke StarTool FDM from ISPF option 6, as a line command from ISPF option 3.4, or with the ISPF TSO command without restrictions.

READY mode

When you enter the first subcommand using ISPF services, StarTool FDM calls an ISPF interface that invokes itself recursively as a full ISPF dialog.

Another TSO command

If an ISPF environment was established by ISPF before calling the other command, there are no restrictions on the use of ISPF services by StarTool FDM as a called command. However, if the ISPF environment was established by the other command processor itself, called commands cannot use ISPF services. If no ISPF services are used before StarTool FDM is entered, this is equivalent to "From READY mode" above.

In the background

If you execute StarTool FDM under IKJEFT01 (the TMP) in a batch job, ISPF services are available but StarTool FDM cannot perform panel prompting, and line commands are not supported except by the APPLY command.

The BROWSE, EDIT, ISPF, ISPMODE, ISPXEQ, MEMLIST, OUTCOPY and CONTROL (with DSN or SYSOUT operands) subcommands can trigger the recursive StarTool FDM dialog invocation. The OUTCOPY and CONTROL subcommands do not require ISPF services; however, since they open a DCB for extended periods of time, an ISPF environment is established so that ISPF services can be used later.

Chapter 5: Dialog Processing

In ISPMODE, several processing functions are available to be used independently or together to give maximum productivity. As a part of this productivity enhancement, these functions have a common look and feel. This is evident in the HELP panels, the CUA panels, the OPTIONS feature, the Menu system, the user command panels and the common features to manipulate StarTool FDM displays. CUA panels are documented later in this section. The following paragraphs describe several of the common features.

Most functions provide more than one display to provide more than one mix of information or one style of display. You can select these displays using the LEFT and RIGHT commands or their related PF keys.

All functions but CALC, HEX, STATUS, PBROWSE and LOG allow for line commands. When line commands are allowed, the functions provide a command field and a data/message field. The command field is four characters long to provide room for a selection of commands as well as related block commands. Use the data/message field to enter operands allowed by some line commands. It is also used as a feedback area. The PEDIT processor does not provide a data/message field.

For any of the table or log panels, the command input field is 48 characters wide on an 80-character screen. If a command you are entering exceeds this input field, enter the first 47 (or less) characters followed by a + to indicate more data is required; a continuation panel with more room for operand entry displays.

In StarTool FDM each function is available concurrently in parallel. This means that as you create functions (logs, MEMLIST tables, LISTA tables, LISTC/LISTF tables, and so forth) you can move from one to another without losing the current function. When you select a function, you return to it in an intelligent fashion. For example, in the log you are at the top of the log for the last command. If you previously repositioned the log and you select it again with no log output additions, you are at the same point again. Similarly, with the table panels, you are positioned at where you left the table unless one or more line commands acted on the table. In that case, positioning is for the last selected item.

USING CLISTS IN ISPMODE

There are multiple ways to start a CLIST in StarTool FDM. One CLIST category can be referred to as SELECT, or external, because you invoke it through the ISPF SELECT service. This type of CLIST can only interact with StarTool FDM through ISPF shared variables or passed variables.

Following are three examples of starting a CLIST or REXX EXEC through the SELECT service:

```
TSO %MYCLIST PARM1
TSO EXEC MYLIB.CLIST(MYCLIST) 'PARM1'
CMD(%MYCLIST PARM1) from a StarTool FDM dynamic command or
user extension panel
```

The other major CLIST category is called internal because it runs under StarTool FDM control and is supported by StarTool FDM directly. This type of CLIST support is covered later in this chapter. A REXX EXEC does not work because REXX cannot address the TSO input stack and REXX is not aware of command processor subcommands.

Following are two examples of starting an internal CLIST from StarTool FDM:

```
%MYCLIST PARM1
EXEC MYLIB.CLIST(MYCLIST) 'PARM1'
```

Use the internal CLIST support in StarTool FDM for different purposes. For example, you can use these CLISTs to:

- demonstrate StarTool FDM by executing preset demonstration scripts
- train users on the effective use of StarTool FDM
- provide a simplified interface for unsophisticated users
- help users set up their initial StarTool FDM environment
- implement new services in StarTool FDM and
- for other types of applications requiring user interaction with a good level of control.

Use CLISTs in ISPMODE interactively and in batch. In batch, any request that requires a prompt panel terminates the CLIST. When run interactively, there is more flexibility because you can respond to the current operating conditions.

The ISPEXEC service of ISPF is not available in these CLISTs because StarTool FDM is executing in subcommand mode. As a consequence, several services are needed to duplicate ISPF service calls from a StarTool FDM environment.

Because a CLIST is run in subcommand mode, all commands in a CLIST are processed as StarTool FDM command line input. ISPF variables used by StarTool FDM are separate from CLIST variables. CLIST statements are processed by the CLIST processor. StarTool FDM is only aware of commands in the CLIST that the CLIST processor determines are not its own.

Chapter 5: Dialog Processing

During CLIST processing verification prompting for data set updates are turned off and the requested updates are performed. This means that a CLIST cannot respond YES or NO to a subcommand like FIXPDS RESET because YES is always assumed. In addition, a StarTool FDM CLIST that invokes a browse or edit session (such as ISPF browse, ISPF edit, PBROWSE, PEDIT, EDITLOG or EDITTBL) cannot interact with the session. The CLIST however, gains control again when you terminate the browse or edit session.

Use several StarTool FDM subcommands in a CLIST to control the ISPMODE environment. These subcommands are DCONTROL, DSETVAR, DGETVAR and PANEL.

- DCONTROL controls how displays and their responses are to be handled. Normally, tables are not displayed during CLIST processing for command input and by default, prompt panels display and their inputs are supplied interactively. DCONTROL can explicitly control how these displays and prompting panels operate.
- Use DSETVAR to set ISPF variables used by StarTool FDM. In most cases, this is for setting messages, prompt panel values, or line commands.
- DGETVAR command retrieves ISPF variables used by StarTool FDM. In most cases, this is used to check the conditions of the current table, the text of the last message, or an expected user response.
- Use the PANEL command to display panels. It displays table panels, but only headings and the last processed row display.

DCONTROL Command Syntax

```
DCONTROL DISPLAY/NONDISPL/ENTER/LOCK/NOEND/ENDOK/SKIP/NOTE/MSG/SETMSG/  
      POP xx yy/UNPOP/ROW nn/NEXT nn/LINES/EXPRESS/WAIT nn/DELETE /  
AUTOLOCK nn ss/AUTONOND nn/AUTOEND nn
```

Note You can enter only a single keyword on a DCONTROL statement that uses number operands. You can enter multiple DCONTROL commands.

DCONTROL Operands

- **DISPLAY** Causes the current table panel to display. You can enter commands unless you also specify LOCK. The command in ZCMD after the display is processed unless you specify SKIP.
- **NONDISPL** Causes the next panel to process but not display. An END command is simulated unless you also specify ENTER. This is used to skip the display of an expected prompt panel.
- **ENTER** Used with NONDISPL to simulate ENTER.
- **LOCK** Causes the next display to not allow user input; ENTER is simulated. Use this option with the DISPLAY operand or before a PANEL command. Do not use it with NONDISPL or ENTER operands.
- **NOEND** Causes an END response to terminate the CLIST. This is the default mode. In a MENU panel, END backs up one level and when no levels are left, the CLIST terminates.
- **ENDOK** Causes an END in the next displayed table or panel to be accepted. If any special processing is needed, the CLIST must perform it. **Note:** Do not use this before a PANEL command with a SINGLE operand.
- **SKIP** Used with DISPLAY to ignore the next command entered from a table display. An interactive user can view, scroll, and use the CUA features of a table panel without being able to enter commands. When the user presses ENTER, CLIST processing resumes.
- **NOTE** Causes the next message to be a PDS#102M message that provides a multi-line message with no alarm (usually an informational message). Use this with DSETVAR for NOTE1-NOTE6 and either the DISPLAY operand, a SETMSG command or the PANEL command.
- **MSG** Causes the next message to be a PDS#101A message that provides a single-line message with an alarm (usually a error message). Use this with DSETVAR for NOTE1-NOTE6 and either the DISPLAY operand, a SETMSG command or the PANEL command.
- **SETMSG** Calls the ISPF SETMSG service to add a message for the next display. Use this with a NOTE or MSG operand.
- **POP** Causes the next panel to display in a window. The vertical row position is set by the first number operand. The horizontal column position is set by the second number operand. Insure that a UNPOP is used before resuming normal displays. POP

Chapter 5: Dialog Processing

requests can be stacked and the numbers can be negative (they are relative to the top left corner of the current display).

- **UNPOP** Causes the current window to be removed. The number of UNPOPs should match the number of POPs in order to resume normal displays. When a CLIST terminates, an UNPOP is performed for every unmatched POP remaining.
- **ROW** Causes the current table to be positioned to the row number specified by the number operand. This row number becomes the top of the display until another ROW or NEXT is used.
- **NEXT** Causes the current table to change its position by the number of rows specified by the number operand. This value can be a negative number to move the position up a number of rows. This row number becomes the top of the display until another ROW or NEXT is used. The default for the number operand is 1.
- **LINES** Causes line command processing to begin in a table panel. Processing starts with the first line command. Additional line commands are set pending and are processed one at a time when control returns to this function.
- **EXPRESS** Causes line command processing to begin in a table panel. Processing starts with the first line command and all line commands are in express mode (transitions back to this function are not required to process all line commands).
- **WAIT** Causes the CLIST to wait for the number of seconds specified by the number operand, or a default of 20 seconds if a number is not entered.
- **DELETE** Causes the current LOG line to be deleted. This is only supported for a LOG table.
- **AUTOLOCK** Causes the next series of prompt panels to display without allowing user input. ENTER is simulated. The first number operand is the count of prompt panels to be processed in this manner. The second number operand is the number of seconds the display is shown to be processing with normal processing. Do not use this with other operands that use number operands
- **AUTONOND** Causes the next series of prompt panels to be processed without being displayed. ENTER is simulated. The number operand is the count of prompt panels to be processed in this manner. Do not use this with other operands that use number operands. It cannot be used with AUTOLOCK.
- **AUTOEND** Causes the next series of prompt panels to be processed

without being displayed. END is simulated. The number operand is the count of prompt panels to be processed in this manner. Do not use this with other operands that use number operands. It cannot be used with AUTOLOCK or AUTONOND.

DSETVAR Command Syntax

DSETVAR internal-variable/VAR(ISPF variable) 'character string'

DSETVAR Operands

- **internal variable** is one of the following:
 - PDSZSEL** command variable from prompt panels
 - MSG** message text used by PDS#101A msgid
 - NOTE1-NOTE6** message text for lines 1-6 of message PDS#102M
 - OPT** current line command input field
 - DATA** current line data/msg input field
- **VAR(name)** name of an ISPF variable in the function pool of StarTool FDM. The variable can be found in various ISPF panels used by StarTool FDM.
- **'character string'** the character string to be placed in the desired variable. If the string has imbedded blanks, it must be surrounded by single or double quotes. If the string has imbedded single quotes, surround the string with double quotes.

Message PDS#102M

Message PDS#102M contains six lines of text, but depending on the length of the text lines, fewer lines may display, as shown below:

Characters per line	Displayed message lines
1-31	6
32-50	5
51-76	4

Chapter 5: Dialog Processing

You can also reduce the lines displayed by setting the lines to be displayed and setting all others to blanks.

Note If a message is part of a popup panel that is positioned near the right edge, message lines wrap to additional lines. Leave room on the right for the message to display.

DSETVAR and DGETVAR

Save variable information for use by several related CLISTs with DSETVAR to create implicit variables in the StarTool FDM function pool. The StarTool FDM function pool acts like a shared pool for CLISTs that are called by one another. Then, any CLIST can use DGETVAR to retrieve the saved variable. You can create implicit variables in panels displayed by a CLIST.

DGETVAR Command Syntax

DGETVAR internal-variable/VAR(ISPF variable) CLIST-variable

DGETVAR Operands

- **internal variable** is one of the following:
 - ZCMD** command line input variable
 - PDSZSEL** command variable from prompt panels
 - MSG** message text used by PDS#101A msgid
 - OPT** current line command input field
 - SIZE** number of rows in the current table
 - DATA** current line data/msg input field
 - FUNC** character assigned to current function, as follows:
 - A** for LISTA
 - C** for LISTC/F
 - F** for FLOATING CALC
 - G** for CSECT
 - H** for HEX CALC
 - K** for CMDTBL
 - L** for PROFMAN
 - M** for MEMLIST
 - R** for VMAP
 - S** for LOG
 - T** for WORKPAD
 - V** for LISTV
 - X** for CAX
 - Z** for ZAP
- **VAR(name)** Name of an ISPF variable in the function pool of StarTool FDM. The variable can be found in various ISPF panels used by StarTool FDM.
- **CLIST-variable** Name of the CLIST variable that is to be initialized with the value of the ISPF variable.

PANEL Command Syntax

```
PANEL panel-name blank/NOP/SINGLE/MENU COPY NONDISPL
```

PANEL Operands

- **panel-name** Name of an ISPF panel to display. This can be a StarTool FDM panel or any other panel accessible from the ISPF environment.
- **blank** With no NOP, SINGLE or MENU operand, the panel is processed as part of the OPTION system. The OPTION system supports panel stacking or command execution from syntax and entry assist panels based on the contents of PDSZSEL.
- **NOP** Initializes the command return variable to NOP. Use this with panels that do not set the command return variable PDSZSEL.
- **SINGLE** Causes just a single panel to display and the command return variable PDSZSEL to be ignored. END commands do not terminate the CLIST. On return from the display, normal CLIST processing resumes. Any additional support must be provided in the CLIST.
- **MENU** Indicates the panel is to be processed as part of the MENU system. This is ignored if SINGLE is specified. This is similar to the blank option explained above except that hierarchical processing is also requested.
- **COPY** Indicates the PDSZSEL value is to be copied into ZCMD before displaying the panel. ZCMD is always cleared by the PANEL command before any processing. This allows ZCMD to be set after it is cleared. After this, PDSZSEL is initialized to a default value that would be NOP for NOP or ? otherwise.
- **NONDISPL** Indicates that the panel is to be processed but not displayed. ENTER is simulated. This overrides the NONDISPL option of the DCONTROL command.

CLIST Processing Guidelines

Start your CLISTs with CONTROL NOCAPS if you want upper and lower case message text.

The DCONTROL NOEND default permits you to exit a CLIST by entering END from any display. To control END processing, use DCONTROL ENDOK before each display.

Before starting a CLIST, StarTool FDM is placed in the Log. This is also true if the CLIST is started from another CLIST. It is impossible to tell from which function a CLIST was started.

Take into account automatic EDITLOG displays. When in EDITLOG, the CLIST has no control of the user input. Automatic EDITLOG can be on by default but can be turned off by setting variable PDSEDLG to NO using DSETVAR.

Position to a function and display a panel before processing but do not use END for function switching. Switch back to a function with a pending line command explicitly. Do not use the PEND command or StarTool FDM may select the incorrect pending function.

Check to see if the table was built as expected to avoid unexpected operand prompts. Check the status of a table by using the following ISPF variables in StarTool FDM for one of the status values (I for Inactive, A for Active and P for Pending line commands):

- PDSSTA** for LISTA
- PDSSTB** for PBROWSE
- PDSSTC** for LISTC/F
- PDSSTE** for PEDIT
- PDSSTF** for CALC
- PDSSTI** for CMDTBL
- PDSSTL** for LOG
- PDSSTM** for MEMLIST
- PDSSTN** for HEX CALC
- PDSSTP** for PROFMAN
- PDSSTQ** for VSAM services
- PDSSTR** for VMAP
- PDSSTS** for CSECT
- PDSSTT** for WORKPAD
- PDSSTV** for LISTV
- PDSSTX** for CAX
- PDSSTZ** for ZAP

The user can see the progress of the CLIST by using DCONT DISPLAY LOCK followed by a DCONT WAIT 5.

Chapter 5: Dialog Processing

To examine output with scrolling, use DCONT DISPLAY SKIP. Then, press Enter and continue with CLIST processing.

You can also set up a multi-line message with a display, as shown below:

```
CONTROL NOCAPS                ; /* allow lower case messages
DSET NOTE1 'Message line 1'    ; /* format a six line
DSET NOTE2 'Message line 2'    ; /* message of some sort
DSET NOTE3 'Message line 3'
DSET NOTE4 'Message line 4'
DSET NOTE5 'Message line 5'
DSET NOTE6 'Message line 6'
DCONT DISPLAY LOCK NOTE        ; /* display table and message
DCONT WAIT 10                  ; /* wait 10 seconds
```

To view output as long as you want, display a pop-up panel along with the output that instructs you to press Enter. StarTool FDM supplies the PDS8ENTR panel for this purpose:

```
_____ Press ENTER
         to continue
```

Messages are placed in a box just below this panel. This is an example of panel PDS8ENTR:

```
DCONT DISPLAY LOCK
DSET NOTE1 'Message line 1' ; /* format only a 4 line */
DSET NOTE2 'Message line 2' ; /* message of some sort */
DSET NOTE3 'Message line 3'
DSET NOTE4 'Message line 4'
DSET NOTE5 '                ' ; /*     these will not     */
DSET NOTE6 '                ' ; /*     be displayed     */
DCONT NOTE NOEND POP 11 40 ; /* right below off center */
PANEL PDS8ENTR
DCONT UNPOP                    ; /* remove panel after ENTER */
```

Another way is to give you control of the panel. With DCONT DISPLAY SKIP, you can use any ISPF commands such as UP or DOWN to view the table. You can set it up so you can have several tries or you can check ZCMD for F and RFIND commands and pass them on to StarTool FDM. For example:

```
DO I=1 to 5                /* allow 5 tries
LOOP: -
DCONT DISPLAY SKIP NOEND  /* allow viewing
DGET ZCMD IN              /* get user input
CK = &SUBSTR(1:2,&STR(IN))
IF CK=F THEN DO          /* is it a find
&IN                      /* yes do it
GOTO LOOP                /* another try
END
IF CK=RF THEN DO        /* is it a RFIND
&IN                      /* yes do it
GOTO LOOP                /* another try
END
END                      /* end of do loop
```

To display an informational panel or panel to get dialog variables from the user that are needed by the CLIST, use the PANEL with the SINGLE operand. If you are not looking for a response from the user, use DCONT LOCK before the PANEL command.

For example:

```
DCONT POP 2 10 LOCK      /* put at center near top
PANEL INTRO SINGLE      /* panel has WINDOW(60 12)
DCONT WAIT 30           /* display for 30 sec
DCONT UNPOP             /* remove panel
```

This is an example of a CLIST to prompt for a test data set name:

```
DCONT POP 2 10
PANEL GETDSN SINGLE     /* input field is TESTDSN
DCONT UNPOP
DGET VAR(TESTDSN) TSTDSN /* copy DSN into TSTDSN
```

Chapter 5: Dialog Processing

This is an example of a CLIST demonstrating the MENU system:

```
DSET NOTE1 'message'      ;/* set description
DSET PDSZSEL 'FIXPDS'     ; /* set PDSZSEL for PANEL
DCONT NOTE LOCK
/* PDSZSEL will be copied to ZCMD
PANEL PDSMENU SINGLE COPY ; /* display with ZCMD set
DCONT WAIT 10             ;/* wait 10 seconds
DSET NOTE1 'message'     ;/* set for next display
DCONT NOTE LOCK
PANEL PDSOPFX SINGLE      ;/* show panel
DCONT WAIT 10
DSET VAR(PDS#BLAD) '20'  ;/* set to add 20 directory blocks
DCONT NOTE LOCK
PANEL PDSOPFX SINGLE      ;/* show the user the change
DCONT WAIT 10
/* CONTROL NOPROMPT is forced in CLIST mode
PANEL PDSOPFX NONDISPL    ;/* panel creates FIXPDS command
DCONT DISPLAY LOCK       ;/* display results
DCONT WAIT 10
```

Check that StarTool FDM is processing the function expected by using DGET for FUNC to get the letter corresponding to the current table.

The following sample panels are supplied for CLIST processing:

- PDS8SAM6 is a 6 row by 31 character blank panel with a HIT enter at the bottom. Each panel should have at least one input field.
- PDS8FUNC is a simulated function pulldown menu for demonstrating the uses of action bars.

This is an example of using this panel:

```
DCONT LOCK POP 0 -1      ;/* it goes just under the FUNCTION
PANEL PDS8FUNC SINGLE
DCONT WAIT 10
DCONT LOCK POP 6 -4      ;/* it goes just under the LISTA
PANEL PDS9LTAP SINGLE    ;/* normal help panel for LISTA
DCONT WAIT 10
DCONT UNPOP
DCONT UNPOP              ;/* need 2 unpop for the 2 pops
```

The tables built in ISPMODE can be output to members of data sets for additional processing. For example, use the following CLIST segment to copy the LOG table (or any other table like MEMLIST or LISTC) to a member of a PDS:

```
/* OUTPUT A TABLE TO A MEMBER OF A DATA SET WITH NO PROMPTING */
DSET VAR(PDSPXDSN) "'HIGHLEV.LIB.CNTL(DUMMMEM)'"
DSET VAR(PDSPDISP) 'S' ; /* DATA SET DISPOSITION */
DSET VAR(PDSPHDRI) 'YES' ; /* SUPPRESS TABLE HEADER */
DSET VAR(PDSPLAST) 'NO' ; /* LAST COMMAND ONLY */
DSET VAR(PDSPCLAS) '' ; /* OUTPUT TO A DATA SET */
PAN PDSVEXL SINGLE NONDISPL ; /* JUST SET THE VARIABLES */
OUTPUT TESTOUT2 ; /* ACTUAL OUTPUT MEMBER */
```

There is no simple way to display a table panel with a sample command in the command input field; however, you can display the panel without the table information by using the PANEL command. The sample command is not executed but the same command can be provided by the CLIST after the display. See the following example which uses the normal LOG display panel for source data sets:

```
DSET PDSZSEL 'USAGE ALL' ;/* set PDSZCMD for later
DCONT LOCK
/* PDSZSEL will be copied to ZCMD
PANEL PDSPNA0 SINGLE COPY ; /* display with ZCMD set
DCONT WAIT 10 ;/* wait 10 seconds
USAGE ALL ;/* do actual command
DCONT DISPLAY LOCK ;/* show results
DCONT WAIT 10 ;/* wait 10 seconds
```

Line command processing is also possible with these services.

- Use DCONT ROW 1 to position a table to the first row
- Use DSET to set the line command field
- Use DCONT NEXT to skip down a row
- After setting a number of rows enter a DCONT ROW 1 to go to the top, followed by DCONT DISPLAY LOCK to show the user the line commands before execution
- DCONT LINES executes the line commands

Chapter 5: Dialog Processing

Check execution progress and respond to the results since you have little control over prompt panels that may result. Plan ahead to tell the user what to do, or do whatever is necessary to avoid prompting. DGET SIZE returns the size of a table. Use this size to control loops that set a number of rows. It can also use the FUNC variable and PDSSTx status variables to track commands as they execute and determine the next continuation action. Use DCONT EXPRESS to run through line commands without any required actions by the CLIST until they are all processed.

This is an example of a CLIST demonstrating LISTC line commands:

```
/* assume user is at listc display and starts CLIST
/* the CLIST will do INFO line command for all VSAM data sets
DGET FUNC STAT          ;/* check function
IF &STAT ^= C THEN EXIT ;/* something wrong
DGET SIZE TSIZE         ; /* get size
IF &TSIZE = 0 THEN EXIT ;/* something wrong
DO I=1 TO &TSIZE        ; /* loop through all rows
    DCONT ROW &I       ; /* each row
    DGET VAR(PDSCVOL) V ; /* volume name field
    IF &STR(&V)=&STR(*CLUST) THEN DO ;/* a VSAM file?
        DSET OPT "INFO" ;/* yes, set INFO line command
    END
END                      ;/* end do loop
DCONT ROW 1              ; /* at top
DCONT DISPLAY LOCK      ;/* show user the plan
DCONT WAIT 5            ;/* wait a little
DCONT EXPRESS           ;/* start displays
/* the user will see the INFO display and press ENTER to continue
/* CLIST resumes after all INFO displays have been done
```

This is an example of a CLIST demonstrating MEMLIST line commands:

```
DGET VAR(PDSSTM) STAT   ; /* check current status
IF &STAT=P THEN EXIT    ; /* busy, skip it
MEMLIST :               ; /* build memlist table
DGET VAR(PDSSTM) STAT   ; /* check current status
IF &STAT=I THEN EXIT    ; /* can not build the table
DGET FUNC STAT         ; /* check function
IF &STAT ^= M THEN ML   ; /* not at memlist
DGET SIZE TSIZE        ; /* get size
IF &TSIZE < 5 THEN EXIT ; /* need 5 members for demo
DCONT ROW 1            ; /* start at row one
DSET OPT 'INFO'        ; /* info on line 1
DCONT NEXT
```

```
DSET OPT 'REPR'           ; /* REPRO
DSET DATA 'TESTNAME'    ; /*      TO TESTNAME
DCONT NEXT
DSET OPT 'ALIA'          ; /* add alias
DSET DATA 'ALTNAME '    ; /*      of ALTNAME
DCONT NEXT
DSET OPT 'VER '          ; /* verify
DSET DATA '*VERIFY*'    ; /* show message
DCONT NEXT
DSET OPT 'BROW'          ; /* browse
DSET DATA '*BROWSE*'    ; /* show message
DCONT ROW 1 DISPLAY LOCK ; /* at top and display
DCONT WAIT 3             ; /* wait a little
DCONT LINES              ; /* start execution
DCONT DISPLAY LOCK       ; /* results of INFO
DCONT WAIT 3             ; /* wait a little
ML                        ; /* do next line command
DGET FUNC STAT           ; /* check function
IF &STAT ^= M THEN DO    ; /* did repro fail
    DCONT DISPLAY LOCK   ; /* display error in repro
    DCONT WAIT 3         ; /* wait a little
    ML                    ; /* return to ML for next
END
DGET FUNC STAT           ; /* check function
IF &STAT ^= M THEN DO    ; /* did alias fail
    DCONT DISPLAY LOCK   ; /* display error in repro
    DCONT WAIT 3         ; /* wait a little
    ML                    ; /* return to ML for next
END
DCONT DISPLAY LOCK       ; /* results of VERIFY
DCONT WAIT 3             ; /* wait a little
DGET FUNC STAT           ; /* check function
IF &STAT = S THEN ML     ; /* should be at log
/* browse will be shown until user ends
DCONT DISPLAY LOCK       ; /* show memlist table
DCONT WAIT 3             ; /* wait a little
```

BATCH PROCESSING

6

It surprises most users that batch operation of StarTool FDM is just as flexible and easy to use as StarTool FDM in line mode from a terminal. The complicated maneuvers that you learn to take for granted are just as easily accomplished in a batch job. The big difference is that you must anticipate all the steps. In batch, you cannot evaluate the results of the first step before beginning the execution of the following steps. The results of all StarTool FDM steps are displayed in batch, however, because StarTool FDM always uses the PUTLINE-GETLINE TSO interface in its operation. With a little practice, you can design incisive and efficient batch jobs to perform many tasks.

You may want to design a batch job under circumstances when you would rather not wait at a terminal for some long execution. Examples might be adding ISPF statistics for a lot of libraries. Cloning your system libraries or other libraries through repeated execution of COPY and CHANGE subcommands.

CLISTS that execute StarTool FDM subcommands may be long in running. It may be easier to execute them in a batch job, rather than fussing with the same process at a terminal.

Several facilities are useful in this environment.

- You can place the CONDEND (conditional END) subcommand after a critical StarTool FDM subcommand (see example 2 below) to terminate StarTool FDM if a warning or error message is generated by that subcommand.
- To capture entered subcommands and test their syntax before creating a batch JOB, enter CONTROL TESTSYNTAX. ISPMODE facilities still function as well as the CHANGE, CONTROL, QUIT and END subcommands.
- To suppress informational messages, enter CONTROL NOINFO.
- To suppress subcommand echoing in the session log, enter CONTROL NOCOMMAND.
- To suppress all messages (except error messages), enter CONTROL NOPUTLINE
- To capture the session output in a RECFM=FB,LRECL=80 data set, enter a subcommand similar to CONTROL DSN(data.set(memnam)). You can also add a disposition parameter like OLD, MOD, SHR or NEW.

Chapter 6: Batch Processing

It is often effective to generate input for StarTool FDM through some process that formats calls to a CLIST as shown in the following:

```
STARTOOL 'SYS1.PROCLIB'  
%CLISTP 'DATA.SET.TO.MANIPULATE'  
%CLISTP 'ANOTHER.DATA.SET'  
. . .
```

CLISTP contains a StarTool FDM subcommand to CHANGE to the data set and additional subcommands to manipulate the data set as needed. Any data encountered under CLIST mode (from an implied CLIST call or the EXEC subcommand) is assumed to be StarTool FDM subcommands or CLIST commands. If they conflict because their names are the same, you can use an abbreviated or alternate name to reference the StarTool FDM subcommand. For example, to reference IF and CONTROL subcommands, use the subcommand names, IFX and CONTRO, respectively.

One sample of JCL suffices for non-ISPMode applications of StarTool FDM in batch mode. The only variations are in the command input under the ddname SYSTSIN, in the program libraries included in a STEPLIB and in the CLIST libraries specified for SYSPROC. There is extra power in batch mode. Ordinary TSO commands can be executed before or after StarTool FDM executions. Also, StarTool FDM can call ordinary TSO commands while executing by using the StarTool FDM TSO subcommand as an interface within the batch job.

An example with JCL to execute StarTool FDM in the background without ISPF facilities follows. This example adds ISPF statistics to all members in several large libraries. This sample is in the CNTL library distributed with StarTool FDM as member PDSBATCH.

```
//TSOBATCH JOB (JOB CARD PARAMETERS)  
//TSOBATX EXEC PGM=IKJEFT01,REGION=3600K,DYNAMNBR=50  
//STEPLIB DD DISP=SHR,DSN=load.library (if applicable)  
// DD DISP=SHR,DSN=load.library2 (if applicable)  
//SYSPROC DD DISP=SHR,DSN=clist.library (if applicable)  
// DD DISP=SHR,DSN=clist.library2 (if applicable)  
//SYSTSPRT DD SYSOUT=* <== OUTPUT OF COMMANDS  
//SYSTSIN DD * <== TSO COMMAND INPUT  
STARTOOL 'SYS1.HELP'  
ATTRIB : ADDSTATS  
C 'SYS1.MACLIB'  
ATTRIB :  
ADDSTATS ID(ESA43)  
END
```

Two other examples of StarTool FDM batch jobs that accomplish not-so-simple tasks follow.

Cloning all the data sets of your system residence pack to another disk pack. Make a copy of the operating system data sets on another disk pack. The new data sets will eventually have the same SYS1 names as the system data sets. This allows you to catalog these data sets to the IPLed system residence volume using the IDCAMS VOL(*****) keyword, so you can IPL either pack later as a system residence volume. (Assume that you have also done other things necessary to IPL that volume, such as copying IPLTEXT to it, etc.)

```
//SYSTSIN DD *
STARTOOL 'SYS1.CMDLIB'
COPY : 'TST1.CMDLIB' NEW VOL(ALTRES) SUMMARY
TSO DELETE 'TST1.CMDLIB' NOSCRATCH
CHANGE 'SYS1.LINKLIB'
COPY : 'TST1.LINKLIB' NEW VOL(ALTRES) SUMMARY
TSO DELETE 'TST1.LINKLIB' NOSCRATCH
CHANGE 'SYS1.LPALIB' COPY : 'TST1.LPALIB' NEW VOL(ALTRES) SUMMARY
TSO DELETE 'TST1.LPALIB' NOSCRATCH
...
      (repeated for all system data sets on the SYSRES pack.)
      (afterwards do LISTF for all TST1 files on the ALTRES)
      ( pack, and rename them - uncataloged - to SYS1.*** )
...
END
```

Tune SYS1.LPALIB by moving some of its modules to a linklist data set to increase the size of available private storage for all address spaces. This involves copying some members off, with all their aliases, and deleting these members from SYS1.LPALIB, with all their aliases only if the copy is successful.

```
//SYSTSIN DD *
STARTOOL 'SYS1.LPALIB'
COPY XYZ* 'SYS2.LINKLIB' ALIAS
CONDEND
DELETE XYZ* ALIAS
END
```

DYNAMIC TSO ENVIRONMENT

For installations with TSO/E Version 2.3 or above, you can invoke StarTool FDM invoke directly in non-TSO environments. This is implemented by the TSO/E Environment Service (IKJTSEV). It can be used as follows:

- Normal background jobs through JCL with PGM=STARTOOL. For an example, see the following example code.
- In a program such as Assembler or COBOL, calling StarTool FDM and passing a single command or invoking StarTool FDM with a data set and passing subcommands through DDNAME SYSTSIN. For an example of how to invoke StarTool FDM from an assembler program in a non-TSO environment, see member SAMPTSOE in the StarTool FDM ASSEMBLE distribution data set.
- In a VTAM application to execute TSO/E CLISTS or REXX execs.
- For installations with MVS/ESA 4.2 or higher, APPC/MVS can be used to establish a link from a personal computer or workstation to TSO/E.

When you initialize StarTool FDM, it checks to see if a TSO environment is available. If TSO services are not available, StarTool FDM invokes IKJTSEV to build and initialize a TSO environment to allow the use of TSO/E programming services outside of the TSO/E TMP and Service Routines.

The TSO/E environment service offers a number of performance benefits. Performance is improved because the TSO/E TMP is not executed. Instead, StarTool FDM invokes TSO/E services and facilities directly. Thus, a given application can be fine tuned to meet the needs of the installation.

When a dynamic TSO environment is initialized, IKJTSEV checks for the presence of DDNAMES SYSTSIN (for subcommands) and SYSTSPRT (for session output). If either of these DDNAMES are not allocated, IKJTSEV dynamically allocates them to DUMMY data sets. A batch program allocates SYSTSIN or SYSTSPRT to disk data sets and dynamically formats input data before calling StarTool FDM. Then, after receiving control back from StarTool FDM, the program analyzes the session output data to determine if functions were performed properly.

The dynamic TSO Environment service has a number of restrictions:

- You cannot initiate Foreground Initiated Background commands. This means if you attempt to submit a data set to JES, you get *IKJ79204I You attempted to run an unsupported function in a dynamic TSO environment.*
- If you attempt to perform authorized functions, you get *IKJ56637I You attempted to run a command, CLIST or REXX exec from an authorized environment. This is not supported under the dynamic TSO environment.*

Because of these restrictions, the following StarTool FDM subcommands are not supported in a dynamic TSO environment.

- SUBMIT subcommand
- LLA subcommand
- FIXPDS RELEASE for a PDSE
- FIXPDS BLK, TRK, CYL or SPACE(nnn)
- COPY with IEBCOPY as the invoked utility
- COMPRESS with IEBCOPY or FDRREORG as the invoked utility

A sample of JCL to execute StarTool FDM in the background in a dynamic TSO environment follows. This sample is in the CNTL library distributed with StarTool FDM as member PDSDYNAM.

```
//TSOBATCH JOB (JOB CARD PARAMETERS)
//TSOBATX EXEC PGM=STARTOOL,REGION=3600K,DYNAMNBR=50,
//          PARM='STARTOOL 'SYS1.PARMLIB''
//STEPLIB DD DISP=SHR,DSN=load.library (if applicable)
//          DD DISP=SHR,DSN=load.library2 (if applicable)
//SYSPROC DD DISP=SHR,DSN=clist.library (if applicable)
//          DD DISP=SHR,DSN=clist.library2 (if applicable)
//SYSTSPRT DD SYSOUT=* <== OUTPUT OF COMMANDS
//SYSTSIN DD * <== TSO COMMAND INPUT
IF : USERID(SER08) THEN(SUBLIST)
ATTRIB *
END
```

ISPMODE IN BATCH

StarTool FDM supports the batch use of ISPMODE facilities if StarTool FDM panel and message data sets are allocated in an ISPF environment. This facility is useful for different applications such as saving LISTC/LISTF, MEMLIST or WORKPAD tables for interactive use or processing multiple data sets in a table with global commands.

To support ISPMODE facilities in batch, several changes were made:

- Prompt panels simulate an END response.
- Syntax errors terminate the batch session.
- Checkpoints are ignored in the log.
- The OUTPUT command produces PUTLINE output by default.

Chapter 6: Batch Processing

- Line commands are not supported except with the APPLY command.

Several ISPMODE facilities are especially useful in a batch environment:

- Global processing can process all data sets in a LISTA/DDNAME, LISTC/LISTF or WORKPAD table.
- Global processing is effective in searching, updating or compressing multiple data sets. In LISTC/LISTF tables, the FIND, REPLACE and COMPRESS commands skip over non-partitioned data sets.
- FIXPDS with its percentage operators to modify data sets (DIRFREE, ADDFREE and RELFREE) is effective at changing multiple data sets using the GLOBAL command.
- Saved MEMLIST and LISTC/LISTF tables allow a batch application to save member or data set names of interest for other types of processing.
- Saved MEMLIST and LISTC/LISTF tables can also be used to contain a list of member or data set names that a batch application is to process.
- Use the EXCLUDE command to trim dialog tables.
- Use the APPLY command to perform the same line command for all table entries.
- Use the SORT command in conjunction with an F or LOCATE command to position to data of interest in a table followed by an X ABOVE command to trim dialog tables.
- Use the TRAP command to capture PUTLINE output from TSO commands in the log.
- Use the MASK command to build LISTC/LISTF tables.
- Use the FILTER command to set up data set filtering defaults for the MASK command.

In a batch application, you can filter data sets by type with global processing by using a variation of the GLOBAL name. This means that you need not trim a LISTC/LISTF table before executing GLOBAL commands.

The following GLOBAL data set type filter names may be used:

- **GLOBAL** process all data sets (except MIGRAT or ARCHIV data sets)
- **GLOBALM** process PDSE or PDS data sets
- **GLOBALE** process PDSE data sets
- **GLOBALO** process PDS data sets
- **GLOBALS** process source PDS or PDSE data sets only (RECFM is not U)
- **GLOBALL** process load PDS or PDSE data sets only (RECFM is U)
- **GLOBALQ** process sequential or direct data sets only
- **GLOBALN** process non-VSAM data sets only
- **GLOBALV** process VSAM clusters only
- **GLOBALT** process all data sets marked with *TAG* in the DATA/MSG field

A sample of JCL to execute StarTool FDM in the background with ISPF facilities follows. This sample is in the CNTL library distributed with StarTool FDM as member PDSISPF.

```
//TSOBATCH JOB (JOB CARD PARAMETERS)
//TSOBATX EXEC PGM=IKJEFT01,REGION=3600K,DYNAMNBR=50
//STEPLIB DD DISP=SHR,DSN=load.library (if applicable)
// DD DISP=SHR,DSN=load.library2 (if applicable)
//SYSPROC DD DISP=SHR,DSN=clist.library (if applicable)
// DD DISP=SHR,DSN=clist.library2 (if applicable)
//SYSTSPRT DD SYSOUT=* <== OUTPUT OF COMMANDS
//ISPPROF DD DISP=SHR,DSN=profile.library (or a temporary):
//ISPPROF DD DISP=NEW,DSN=*&&ISPF,UNIT=SYSDA,SPACE=(TRK,(5,5,5)),
// DCB=(LRECL=80,BLKSIZE=9120,RECFM=FB)
//ISPPLIB DD DISP=SHR,DSN=panel.library (and concatenations)
//ISPMLIB DD DISP=SHR,DSN=message.library (and concatenations)
//ISPSLIB DD DISP=SHR,DSN=skeleton.library (and concatenations)
//ISPTLIB DD DISP=SHR,DSN=table.library (and concatenations)
//SYSTSIN DD * <== TSO COMMAND INPUT
PROF MSGID PREFIX(userid)
ISPSTART PGM(STARTOOL) PARM(STARTOOL SDS.CNTL) NEWAPPL(ISR)
FILTER NOMIGRAT
MASK SER08.*.DATA
OUTPUT
COMPRESS
END
```

F Command

Every function has a F (FIND) command to locate character strings within the function tables.

The find string is positional (it must be entered just after the F keyword). You normally enter it as a simple string delimited by blanks; however, it can be delimited with quotes. The following show three valid FIND commands:

```
F 'This is the first string' first
F "this is the second string" last
F mydata dsname
```

The ANY keyword is defaulted with a FIND command. This means that all data columns are searched. To search only a specific column of data, enter the name of that data column. You can direct the find search with keywords such as NEXT, FIRST, LAST, PREFIX, SUFFIX and WORD. These keywords have the normal ISPF-implied meanings. If you are dealing with a lot of information, the FIND facility can help zero in on any information.

EXCLUDE Facility

EXCLUDE facilities are available for all function tables. From a log or MEMLIST table, use the EXCLUDE subcommand to trim members from the MEMLIST table. The EXCLUDE subcommand supports the same attribute filtering operands that are available on a MEMLIST subcommand.

From other tables, use the EXCLUDE command to trim all table elements that match a search string (or do not match a search string with the NOT keyword). The EXCLUDE command supports the same operands as are available on the F command with the exception of the directional keywords (NEXT, FIRST and LAST) because EXCLUDE processes an entire table.

SORT Command

Use the SORT command in functions with sorted tables such as CAX, CSECTS, LISTC/LISTF, LISTV, MEMLIST and NUCMAP. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.

L (LOCATE) Command

Use the L (or LOCATE) command in functions with sorted tables. The L command locates data in the current sorted column. A partial string can be supplied to obtain a generic search.

X Command

Another standardized feature is the X command. When you use X with an operand, it trims an entire table. Supported operands are ALL, ABOVE and BELOW where the current position in the table is the trim reference point.

OUTPUT Command

The OUTPUT command is available for all functions to output the current log or table to SYSOUT or a data set.

In batch mode, if you do not enter an operand on the OUTPUT command, the table is output with PUTLINE messages. You can also enter an operand such as =x or F(ddname) after OUTPUT. If you enter =x , output is directed to SYSOUT=X . If you enter F(ddname), output is directed to the named preallocated data set.

CONCEPTS AND FACILITIES

7

StarTool FDM has a different look-and-feel when compared with ISPF panels, commands and member lists because of its direct approach in managing data sets and its indirect method of managing members.

StarTool FDM is command-driven much like ISPF Edit and Browse. StarTool FDM supports line commands that you enter on the left side of an individual table line (in the CMD field) to operate on a single table entry, and primary commands that you enter on the top line (or OPTION line) of a panel. Primary commands are:

function commands	control different ISPF tables; these functions support their own commands
subcommands	request operations that are available in all modes of StarTool FDM use
commands	request StarTool FDM operations that are available in ISPMODE only
dynamic commands	request StarTool FDM, TSO or ISPF facilities; can be programmed from a panel
ISPF commands	request ISPF services such as HELP, SPLIT, SWAP, DOWN and UP

StarTool FDM presents an environment with total control over data sets, including data set attributes, allocation and members. This approach saves time used to access data set and member functions in ISPF which requires the activation of multiple separate applications to perform similar functions. StarTool FDM provides access to all library and member maintenance functions from a single environment.

All ISPF features of editing, browsing, renaming, copying and deleting members are accessible from a single member list, which can be built on over 90 different criteria. You can select members by partial member names, member contents or member attributes such as ISPF statistics or linkage-edit date for load members. Once you build a member group, StarTool FDM treats it as an object, grouping all of these members together for use by different subcommands. Various functions ranging from

Chapter 7: Concepts and Facilities

the creation of member lists, copying, searching, updating, comparing, browsing, editing or deleting objects is possible. There is complete data set management capability through several utility commands that search catalogs, volumes or TSO allocations to present a list of data sets where you perform maintenance functions. You can save these data set lists across TSO sessions to maintain a working set of commonly accessed data sets. Complimenting these features are a group of utilities unique to StarTool FDM.

StarTool FDM provides on-line facilities to act on information obtained from IBM batch utilities such as module linkage-edit dates, CSECT maps and module update history. You can perform many common linkage-editor functions interactively. These functions include assigning member aliases, altering member attributes or creating batch JCL to relink multiple modules. Copying members is another StarTool FDM strength. You can copy load members to a data set with a smaller blocksize, copy members into the same library with a new name, and copy PDS members to a sequential data set. You can also convert CLISTS from fixed format to variable format (and vice-versa).

StarTool FDM provides data validation and recovery facilities that include restoring a deleted source or load member from a partitioned data set, verifying a data set or individual members and expanding the allocation (either directory blocks or secondary space) without disrupting normal processing. These features are supplemented by an audit trail that tracks all issued subcommands and their results in a log that you can view, scroll through, edit, print and save.

LLA MANAGED LIBRARIES

Use the LLA (library lookaside) address space to manage all linklist libraries and other load libraries for JOBLIB or STEPLIB use.

When StarTool FDM references a member in an LLA managed library by a LOAD or BLDL service, directory entries maintained by LLA are used instead of the directory entries on disk. Because StarTool FDM uses LLA-cached directory entries, these entries may not be synchronized with the directory entries on disk due to member directory attribute changes or member updates.

You can use the StarTool FDM LLA subcommand to selectively refresh LLA directory entries. Use the SYNC operand to synchronize members in linklist libraries because SYNC performs a conditional LLA refresh for members whose disk directory entry is not the same as their LLA directory entry. Use SYNC to update only changed members in the LLA directory; thus members are not staged needlessly.

StarTool FDM provides several types of information for LLA managed libraries:

- When you change to a library (or use the USAGE subcommand), StarTool FDM informs you if the library is LLA managed.
- When you enter a subcommand that updates members, StarTool FDM reminds you to refresh updated members if the library is LLA managed.
- StarTool FDM informs you if you reference a member that was deleted but is still in the LLA directory. Display the attributes of these members with the ATTRIB subcommand.

To determine the synchronization status of members in linklist libraries that are LLA managed, use the VERIFY subcommand. VERIFY informs you of the following conditions:

- A member was added on disk but is not in the LLA directory.
- A member was updated on disk but the LLA directory entry still points to the previous version of the module.
- A member has an undetermined LLA status because of a member in a TASKLIB data set or in a higher linklist data set.

LLA and Deleted Members

Use the LLA subcommand as a line command in a MEMLIST table. If you are planning to rename members of an LLA controlled data set, set Remove renamed members in SETML to NO. Then, after you rename a member with the REN line command, enter a LLA line command on both the old and new member names to inform LLA of their status changes. After the LLA subcommands finish, remove the old member name with an X line command.

If you are planning to delete members of an LLA controlled data set, set Remove deleted members in SETML to NO. Then, after you delete a member with the DEL line command, enter an LLA line command on the deleted member name to inform LLA that the member was deleted from the data set. After the LLA subcommand finishes, remove the deleted member with an X line command.

USING COMPUTER RESOURCES

StarTool FDM reduces the overhead associated with managing sequential, direct, partitioned, partitioned extended and VSAM data sets. Users can perform all data management functions within a single environment established and maintained by StarTool FDM.

Chapter 7: Concepts and Facilities

Unlike ISPF, users are not required to jump to different ISPF applications to perform member and data set functions within the StarTool FDM environment. StarTool FDM uses its own internal access methods and data management utilities to perform user requests. StarTool FDM offers one-step solutions to reduce system overhead as compared with normal ISPF processing.

The benefits of using StarTool FDM include reduced paging, fewer input/output operations and lower CPU usage associated with managing data as compared with ISPF or other data management products. In most TSO/ISPF enterprises, StarTool FDM offers a substantial saving and helps defer the need to upgrade hardware.

Paging and Swapping Relief

StarTool FDM is a 900K reentrant program that resides in the MVS LPA allowing all users to share a single copy of the program and reduce the amount of below the 16 Megabyte private region requirements for all TSO users. This also reduces the overhead required for program fetch, virtual storage and swapping activity for all users.

ISPF requires the activation of multiple applications to copy, browse, edit and rename data sets and members. Users must often split their ISPF sessions to perform a simple task. StarTool FDM combines all data management functions into a single workbench environment. Its multiple data set session capabilities reduce the number of ISPF, TSO or PIE/TSO sessions required.

Reducing CPU and I/O

StarTool FDM uses EXCP full-track channel programs to input most of the data it accesses and operates faster and more efficiently than ISPF. StarTool FDM also internally caches up to nine disk tracks read during a subcommand, reducing the number of input operations required to read a PDS directory or multiple members by up to 50%. Installations will see a noticeable reduction in disk I/O operations using this built-in feature. Because of the reduction in I/O and paging activity achieved by StarTool FDM, CPU consumption is directly reduced.

The workbench concept of StarTool FDM allows users to execute all member and data set functions without jumping to other ISPF applications. StarTool FDM is the driver for the data management functions thereby reducing I/O requirements since StarTool FDM manages all member activity from a single member list to browse, edit, rename, restore, delete and copy members. Likewise, from a single customized data set list, a set of global commands are available to search or replace characters strings within groups of members and groups of data sets.

Reducing VTOC Contention

Users who frequently use the ISPF DSLIST (ISPF option 3.4) application benefit by using StarTool FDM because it allows users to build, save and recall customized lists of data sets containing multiple high level qualifiers from catalog and volume searches. Since users save and recall the data set lists, they need not use the DSLIST application to build new data set lists for functions like editing, browsing, renaming or deleting data sets.

Chapter 7: Concepts and Facilities

EXAMPLES

8

StarTool FDM has many subcommands; however, there are alternative methods to perform different tasks. In this chapter, not only are there suggestions made for alternatives and options, but examples are presented to show how subcommands are related to each other.

You can chain multiple menu levels by separating the levels with periods. For example, M.10.2 is the same as entering M (for MENU), followed by option 10 (for search) and finally option 2. One is the short-cut method; the other the standard step-by-step method. The difference between these two methods is the effect that the END key has on the panel hierarchy. Chained options do not back up in a panel hierarchy.

In the following table, different tasks are identified followed with a page reference and the minimum interface level associated with that task. If a task is described with a minimum level of standard, users at the other levels need to understand and refer to this material as well. Standard users who want to expand their understanding of StarTool FDM need to explore tasks described at advanced or power levels. Similarly, advanced users need to be aware of power techniques to improve their effectiveness with StarTool FDM.

Task	Reference	Minimum Level
Adding Directory Blocks	<i>page 85</i>	Standard
Adding a Single Extent in Tracks	<i>page 85</i>	Standard
Adding a Single Extent in Cylinders	<i>page 85</i>	Standard
Adding an Entry Point	<i>page 85</i>	Advanced
Adding ZAP IDR Records	<i>page 86</i>	Advanced
Allocating a Data Set	<i>page 86</i>	Standard
Backing Up Multiple Members	<i>page 86</i>	Advanced
Building a Member List	<i>page 87</i>	Standard

Chapter 8: Examples

Task	Reference	Minimum Level
Changing Secondary Allocation	<i>page 89</i>	Standard
Checking Alias Members	<i>page 89</i>	Advanced
Checkpointing the Log	<i>page 90</i>	Standard
Cloning a Member	<i>page 90</i>	Standard
Combining Multiple Members	<i>page 90</i>	Advanced
Comparing Members	<i>page 90</i>	Standard
Comparing Multiple Members	<i>page 91</i>	Power
Compressing Data Sets	<i>page 91</i>	Standard
Compressing Multiple Data Sets	<i>page 92</i>	Advanced
Controlling GO Sessions	<i>page 92</i>	Power
Controlling Multiple Logs	<i>page 92</i>	Power
Converting CLIST Members	<i>page 93</i>	Advanced
Converting Source Members	<i>page 93</i>	Advanced
Copying Members	<i>page 94</i>	Standard
Defining a Dynamic Command	<i>page 94</i>	Advanced
Deleting All Members	<i>page 95</i>	Power
Deleting Multiple Members	<i>page 95</i>	Standard
Disassembling a Member	<i>page 95</i>	Advanced
Displaying External References	<i>page 96</i>	Advanced
Editing Multiple Members	<i>page 96</i>	Standard
Excluding Multiple Members	<i>page 96</i>	Standard
Executing in Background	<i>page 97</i>	Power
Executing in CLIST mode	<i>page 97</i>	Power

Task	Reference	Minimum Level
Formatting Data Set VTOC Information	page 98	Advanced
Formatting Directory Entries	page 98	Advanced
Formatting Member Data	page 98	Standard
Listing Module History	page 99	Standard
Locating Data in Dialog Tables	page 99	Standard
Managing Data Set Lists	page 100	Standard
Managing Member Lists	page 101	Standard
Manipulating Non-Partitioned Data Sets	page 103	Advanced
Migrating Test Modules to Production	page 104	Standard
Modifying AMODE and RMODE	page 105	Standard
Modifying DCB Parameters	page 105	Advanced
Modifying Member Attributes	page 105	Advanced
Modifying Member Statistics	page 106	Standard
Modifying Several Members' Attributes	page 106	Advanced
Obtaining Assistance	page 106	Standard
Processing Multiple Data Sets	page 108	Standard
Reblocking Source Members	page 108	Advanced
Reconstructing Linkage-Edit JCL	page 109	Advanced
Releasing Space in a Data Set	page 109	Standard
Releasing Space in Multiple Data Sets	page 109	Advanced
Renaming Multiple Members	page 110	Power
Repairing a CSECT in Multiple Members	page 110	Advanced
Resurrecting deleted members	page 110	Standard

Chapter 8: Examples

Task	Reference	Minimum Level
Saving a LISTC table	page 111	Standard
Saving a MEMLIST table	page 111	Standard
Searching a Data Set	page 112	Standard
Searching for Panel, Message or CLIST Members	page 112	Advanced
Searching for System Modules	page 113	Advanced
Searching Multiple Data Sets	page 113	Standard
Selecting Members	page 113	Standard
Specifying a Member Group	page 115	Standard
Switching Modes	page 116	Advanced
Trimming Dialog Tables	page 117	Standard
Updating a Data Set	page 118	Standard
Updating Multiple Data Sets	page 119	Standard
Validating Data	page 119	Standard
Zapping a CSECT	page 120	Advanced

Adding Directory Blocks

The FIXPDS subcommand dynamically adds directory blocks to a data set. For example, you can add 30 directory blocks to a partitioned data set with any of the following:

- M.FIXPDS option: Add directory blocks ==> 30
- FIXPDS EXPANDDIR(30)
- While in an edit session in a member of the data set, type ADDDIR 30

Note The FIXPDS subcommand checks the area that is occupied by the expanded directory for existing members and prompt you with the names of these members. After you respond with YES, these members are moved to the end of the data set and the new directory blocks are added after the current directory blocks.

Adding a Single Extent in Tracks

If a data set has less than 16 extents, you can add a single secondary extent of any size. You can add 25 tracks to a data set with any of the following:

- M.FIXPDS option: Add disk tracks ==> 25
- FIXPDS ADDTRK(25)
- While in an edit session in a member of the data set, type ADDTRK 25

Adding a Single Extent in Cylinders

If a data set has less than 16 extents, you can add a single secondary extent of any size. You can add 2 cylinders to a data set with any of the following:

- M.FIXPDS option: Add disk cylinders ==> 2
- FIXPDS ADDCYL(2)
- While in an edit session in a member of the data set, enter ADDCYL 2

Adding an Entry Point

Use the ALIAS subcommand to add an entry point to a member. For load members, if the alias name to be assigned matches the name of a external symbol, the new alias is assigned the entry point of that symbol.

Chapter 8: Examples

To add an alias called FOX12 to a module called FOXBASE at external symbol FOXONE, do either of the following:

- Enter ALIAS FOXBASE FOXONE followed by RENAME FOXONE FOX12.
- Enter ALIAS FOXBASE FOX12 followed by ATTRIB FOX12 ENTRY(FOXONE).

Adding ZAP IDR Records

The REPRO subcommand adds space to load members for ZAP IDR records. Each ZAP record contains space for 19 historical entries.

To add an IDR record to load member XMAS, do either of the following:

- M.REPRO option: Member or member group ==> XMAS and
Add ZAP history record ==> YES
- REPRO XMAS ADDZAP

Allocating a Data Set

The MODEL command acts as a front end to the IDCAMS subcommand for VSAM data sets or the CREATE subcommand. In both cases, allocation parameters are presented in a prefilled entry assist panel. Use either of the following:

- M.3.2 option A
- MODEL

Backing Up Multiple Members

You can duplicate multiple members in the same data set with the REPRO subcommand as shown in the example below:

```
REPRO ieb* to(xeb)
```

The original members are not affected; they are just copied. Form the new member names by copying each member name and then placing the TO prefix name over the start of the original member name. Thus, with original member names AA, ABBA and IBM04 with a TO(new) prefix, you generate backup members with names NEW, NEWA and NEW04.

Building a Member List

Use the ML or MEMLIST function to build member list tables. MEMLIST is always additive; that is, each invocation adds to the current MEMLIST unless you specify the RESET keyword.

Partial member lists are often built using member group specifications as follows:

Range	Example: MEMLIST start:end
	AB:C selects all members beginning with AB through all members beginning with C.
	C: selects all members beginning with C through the end of the members.
	F: select all members up through members beginning with F.
	: selects all members.
Pattern	Example: MEMLIST part1/part2
	AB/C selects all members whose names contain AB and C anywhere.
	ABC/ selects all members containing ABC anywhere.
	/ABC selects all members containing ABC anywhere.
	You can use placeholders (% or ?) that match any single character anywhere in a pattern specification.

Chapter 8: Examples

Combination Example: MEMLIST start*end

AB*C selects all members whose names start with AB and contain C at the end of the member name.

ABC* selects all members whose names start with ABC.

*ABC selects all members containing ABC at the end of the member name.

You can use placeholders (% or ?) that match any single character anywhere in a combination specification.

Placeholder Example: MEMLIST start%end

A%?C selects all members with four-character names beginning with A and ending with E.

A% selects all members with two-character names beginning with A.

?F selects all members with two-character names ending with F.

You can use placeholders (% or ?) that match any single character anywhere in a pattern or combination specification.

After the member name position, enter operands to filter members by attributes. For example:

- LAST(*nn*) where *nn* is a number of days to filter member age.
- For source members with ISPF statistics, filter on userid with ID(partial-userid).
- For load members, filter using linkage attributes such as RENT, REUS, AUTH, DC, RMODE24.
- For a complete list of operands, type M.ML or O.ML.
- To refer to all members from the current member list with a subcommand, use = as the member group specification. This makes the current member group equal to the members in the member list before performing the operation.

You can save and recall MEMLIST tables. Use the SAVE and MERGE commands to manage saved member list tables. MEMLIST table names can contain one- to six-alphanumeric characters.

Several MEMLIST short-cut commands are available for source or load libraries: TODAY, WEEK and MONTH. These commands add members that were linked or updated recently (according to ISPF statistics). Another useful command is MINE which adds source members saved with your userid in a member's ISPF statistics. If you want a complete list of members in a MEMLIST table, type ALL or 6.

You can trim member lists with the X line command, the X primary command or the EXCLUDE subcommand. The EXCLUDE subcommand can drop MEMLIST members based on attributes.

The IF subcommand has some additional capabilities for selecting members by attributes as compared to the MEMLIST subcommand. For example, to select members updated by *USERMOD UZ23654*, type:

```
IF : USERMOD(UZ23654) THEN(MEMLIST)
```

You can also use the FIND subcommand to select members by contents. For example, to select members containing the string *ABCX*, type:

```
FIND : 'ABCX' THEN(MEMLIST)
```

THEN(NEWML) could have been used in either of the above two examples to reset any current member list.

Changing Secondary Allocation

If the secondary allocation type or amounts are not optimal, change them to a value of 25 tracks with either of the following:

- M.FIXPDS option: Secondary space type ==> TRK and
Secondary amount ==> 25
- FIXPDS TRK SPACE(25)

Checking Alias Members

If you are processing a MEMLIST table, use the ALIASCHK or 5 command to check alias members. This command fills in the real name of alias members in the ALIASOF field; however, if an alias error is found one of the following is placed in this field:

- *ORPHAN Orphan. No corresponding main member is in the data set.
- =BYTTR Orphan. A main member matches by TTR address but it does not have the same name as in the orphan's directory entry.
- =BYNAME Orphan. The main member noted in the directory entry of the orphan member exists but it does not have the same TTR address.

Chapter 8: Examples

Several StarTool FDM subcommands can process alias members. The COPY, DELETE, DUP, REPRO and SUBLIST subcommands each have an ALIAS keyword to indicate that the member group and all associated members are to be included in the process. Associated members for a main member is its alias members. Associated members for an alias member is the main member and any other aliases of that member. All associated members have the same beginning TTR address.

Checkpointing the Log

When a single StarTool FDM subcommand produces more than a fixed number of output lines (by default, 250 lines), StarTool FDM prompts you with a checkpoint_ panel. You must decide whether or not to continue or terminate the processing subcommand.

- To continue the subcommand, type CONTINUE.
- To terminate the subcommand, press END.
- To find data in the log, type the F command; to find the next string, press RFIND.
- To change the checkpoint interval, type SETLOG.

Cloning a Member

The easiest way to clone a member is with the REPR (REPRO) line command. From a MEMLIST table, type REPR in the CMD field next to the member to be duplicated. Tab over to the DATA/MSG field and type the name for the new member. If a member with that name is already present in the data set, you are prompted about replacing that member.

Combining Multiple Members

Use the COMBINE subcommand to flatten a group of members so that you can manipulate the combined members as a sequential data set in an edit session or for upload/download to a PC.

Use the SEPARATE subcommand to rebuild these members. COMBINE and SEPARATE work together to preserve any ISPF statistics.

Comparing Members

Use the COMPARE subcommand for comparing members. The COMPARE subcommand uses SuperC, COMPAREX or the Yale Compare program to perform the data comparison as specified during StarTool FDM installation.

Note The COMPAREX program is very effective in comparing load members since it compares them logically CSECT by CSECT and it marks differences with underscore characters.

Use a COMP line command in a member list to compare two members in the same data set (type the second member name in the DATA/MSG field). If the members are in different data sets, type O.CO to display a syntax assistance panel so you can specify the comparison data set name (and member).

The syntax assistance panel for COMPARE remembers operands across sessions. If you type the second data set name without a member name for a PDS, the member name defaults to the first member name (this is useful for comparing a member with a backup version in a different data set).

Comparing Multiple Members

The COMPDIR subcommand offers useful possibilities for comparing members in different data sets.

- To select members with differences in two data sets, specify the CHANGE operand.
- To select members found equal in two data sets, specify the NOCHANGE operand.
- To compare associated members (aliases), specify the ALIAS operand.

As an example, to delete redundant members in a data set, perform the following steps:

1. COMPDIR : *other.data.set* NOCHANGE (to select equal members)
2. DELETE * (to delete the selected members)

Compressing Data Sets

The COMPRESS subcommand invokes an external utility program to perform compress functions. Choose the level of reporting detail desired: NOLIST (no details), LIST (all details) or SUMMARY (summary and any error messages). Invoke the COMPRESS subcommand with either of the following:

- M.COMPRESS option: Message level desired ==> SUMMARY
- COMPRESS SUMMARY
- While in an edit session in a member of the data set, type COMPRESS

Compressing Multiple Data Sets

To compress multiple data sets, use the global COMPRESS command. First, build a LISTA/DDNAME, LISTC/LISTF or WORKPAD table containing the data sets to be compressed.

All data sets to be compressed must be partitioned. Eliminate data sets from the table with the X line command. From a LISTC/LISTF table, non-partitioned data sets are bypassed. From the primary command line in a table, type O.COMPRESS to go to the global COMPRESS syntax assist panel.

Controlling GO Sessions

StarTool FDM supports up to nine independent GO sessions (1 through 9). Each GO session can establish its own independent default member group, MEMLIST table, CSECTS table, PBROWSE session, PEDIT session and ZAP table. A CHANGE subcommand in a GO session terminates any of these functions. Other functions, such as LISTA, LISTC, LISTV or WORKPAD are established independently of any GO sessions and are not affected by CHANGE subcommands.

To establish a new GO session, you have several choices. You can type GO *data.set.name* followed by an optional *VOL(volume)* parameter, or from a LISTA/DDNAME, LISTC/LISTF or WORKPAD table, you can type GO as a line command next to a data set to be activated. You can also use the menu primary option 9 (M.9) to activate or terminate GO sessions.

- To toggle between active GO sessions, type GO *
- To rotate through the active GO sessions, type GO **
- To go to any active GO session, type GO *n* (where *n* is the GO session number).
- To select from one of the active GO sessions or activate a new GO session, type STATUS, M.9, 9 or GO.

Each method displays the data set associated with each GO session and its member group.

- To inactivate a GO session, type DROP while that GO session is active. To delete active GO session 3, type DROP 3.

Controlling Multiple Logs

StarTool FDM supports up to nine independent logs (1 through 9). To establish a log, it need only be referenced. For example, LOG 3 goes to log 3, If it was not referenced previously, it would be initialized to a null log.

Use distinct logs for any special purpose. As an example, use log 5 for machine readable data such as SMPGEN or MAP with RELINK outputs while log 7 can be used for data that is referred to over a long period. One example of such long term data might be the output from a VERIFY: subcommand so that individual members can be fixed (in a different log) and validation errors can be referred to later for the next error.

Each log is subject to checkpointing and can be trimmed with the X primary command (use an ALL, ABOVE or BELOW operand). Logs are maintained only for the duration of a StarTool FDM session.

To create a data set or output a log to JES, type the OUTPUT command while viewing the log. To manipulate the entire log in an edit session, type the EDITTBL command while viewing the log. To manipulate the output of the last subcommand in an edit session, type the EDITLOG command from any StarTool FDM function.

Converting CLIST Members

The DUP subcommand supports CLIST conversion if the current data set is F (fixed) with 80 character records and the target data set is V (variable) with 255 character records or vice-versa. This support includes assigning new sequence numbers and splitting long CLIST statements logically as needed.

You can convert CLIST members with either of the following:

- M.DUP option: Member or member group ==> *memgroup* and
Data set or FILE(name) ==> *output.data.set.name*
- DUP *memgroup output.data.set.name*

Converting Source Members

The DUP subcommand supports changing record formats from F (fixed) to V (variable) and vice-versa. This support includes repositioning sequence numbers and padding records with blanks or truncating records as needed.

You can convert source members with either of the following:

- M.DUP option: Member or member group ==> *memgroup* and
Data set or FILE(name) ==> *output.data.set.name*
- DUP *memgroup output.data.set.name*

Copying Members

The COPY subcommand invokes an external utility program to copy members to another data set. You can choose the level of reporting detail desired: NOLIST (no details), LIST (all details) or SUMMARY (summary and any error messages).

For load members, associated members (aliases) are automatically included in a copy. In addition, if COPYMOD reblocking of load members is required due to data set blocksize differences, a COPYMOD operation is requested for the blocksize of the output data set.

Members can be copied selectively. For example:

- REPLACE specifies that existing members in the output data set be replaced.
- EXIST specifies that a member should only be copied to the output data set if a member with that name is already there. Use this parameter in conjunction with the REPLACE operand.
- NOEXIST specifies that a member be copied only to the output data set if a member with that name is not already there.
- EXCLUDE specifies that only members not in the identified member group be copied.

An output data set is created by the COPY subcommand if you specify the NEW keyword. Other parameters for allocation of the data set default to values used for the current data set; however, any of these parameters can be overridden.

To copy selected members into a new data set, enter a subcommand similar to:

```
COPY memgroup new.data.set new
```

Defining a Dynamic Command

Define dynamic primary commands by typing the SETUSER command. From the SETUSER panel, move the cursor to an empty line and type a command name on the left side of the entry area (for example, type MYDATA). Then, position the cursor to the other side of the entry line and type the action (for example, type LISTC B27 to retrieve saved table B27). From the SETUSER panel or from any function in StarTool FDM, enter MYDATA to perform the related function.

To define dynamic line commands, type UT as a line command on any entry in any function. You go to a user panel where you can type a dynamic command in the last part of the user panel. Dynamic line commands are defined in a similar fashion to dynamic primary commands except that command names are limited to four characters.

Different actions are possible:

- Use CMD(*command operands*) to invoke TSO commands or CLISTS.
- Use TRAP(*command operands*) to invoke TSO commands and return their results to the log.
- Use PGM(*pgmname*) to invoke ISPF program applications.
- Use XPANEL(*panname*) to invoke ISPF panel applications.
- Use subcommand operands to invoke StarTool FDM subcommands or functions.

Deleting All Members

When a partitioned data set needs to be reset for reuse, delete all members and compress the data set; however, type FIXPDS RESET for a similar effect. Add the CHECK keyword if you want to check for active edit sessions before the members are deleted. Also, change the number of directory blocks. For example, to set the number of directory blocks to 55, type FIXPDS RESET(55).

Deleting Multiple Members

The DELETE subcommand can also delete associated (alias) members. If you want to delete load members beginning with ABC, enter either of the following to clean up any associated members:

- M.DELETE option: Delete associates also ==> YES
- DELETE *abc** ALIAS

If you enter a subcommand to delete multiple members, the DELETE subcommand displays the members that is affected and prompts for continuation.

Disassembling a Member

Use the DISASM subcommand with operands that limit its output. For example, if load member ABCDOG gets an ABEND at offset X'FC44', type DISASM ABCDOG OFFSET(FC30) to display assembler instructions near the error. Likewise, if you only need data from CSECT BIGDOG in load member ABCDOG, type DISASM ABCDOG MODULE(BIGDOG).

For the menu system, enter:

M.DISASM option: Hexadecimal start offset ==> FC30

Displaying External References

The XREF subcommand lists internal module symbol cross references in a load member. Its output is similar to the output produced by the XREF option of the linkage-editor except that references are provided by name instead of by location. The information provided by the XREF subcommand shows all CSECTS with references to a given symbol and all symbols referenced by a given CSECT. Use XREF as a primary command to process a member group; however, it is usually used as a MEMLIST line command to report on a single member.

Editing Multiple Members

The EDIT subcommand can process a group of members. When you enter an EDIT subcommand with such a group, each member is presented to you in edit (in group order) with no intervening panels.

To modify several members with no displays to the terminal, request an appropriate initial edit macro. As an example for edit macro DOALL, type:

```
EDIT ABC* MACRO(DOALL)
```

Excluding Multiple Members

Use the X command and the EXCLUDE subcommand to trim MEMLIST tables. The X primary command supports an operand (ALL, ABOVE or BELOW) to trim the member table based on the top member shown in the table. Sort the MEMLIST table on some criterion (for example, SORT ID sorts by userid) and position to a member by scrolling with the LOCATE command or the F command and then enter X ABOVE or X BELOW to trim the table. In addition, the X line command or the XX block line command are effective at eliminating undesired members after a SORT.

The EXCLUDE subcommand filters members that use the same attributes as the MEMLIST subcommand (type O.EXCLUDE or M.EXCLUDE to see possible attributes). To exclude members, specify the = member group to examine the members in the current MEMLIST table. Any member group specification is supported. If a member not in the member list is excluded, its status is not changed.

Members can be excluded by attributes (with the IF subcommand) or by contents (with the FIND subcommand) by specifying a THEN(EXCLUDE) operand on the IF or FIND subcommand.

Executing in Background

StarTool FDM subcommand scripts are often set up as batch jobs. In this environment, execute any StarTool FDM subcommand that does not require prompting. Actions that normally prompt with a panel are given a simulated END response, and other actions that normally require a subcommand prompt are allowed to continue without prompting.

The CONDEND subcommand is for batch and CLIST execution. Place this subcommand just after a critical subcommand in a CLIST or batch script to terminate the process with a non-zero return code if any error or warning messages are generated by that critical subcommand.

Several other facilities are useful in this environment.

- To suppress informational messages, type CONTROL NOINFO
- To suppress subcommand echoing in the session log, type CONTROL NOCOMMAND
- To suppress all messages (except error messages), type CONTROL NOPUTLINE
- To capture the session output in a RECFM=FB,LRECL=80 data set, type a subcommand similar to CONTROL DSN(data.set(memnam)). You can also add a disposition parameter like OLD, MOD, SHR or NEW.

Executing in CLIST Mode

StarTool FDM subcommand scripts are often set up in a CLIST or REXX EXEC. This is often performed in a batch process (see [“Executing in Background” on page 97](#)). When you are operating in CLIST mode, StarTool FDM cannot prompt. All input must come from the CLIST itself.

It is often effective to generate input for StarTool FDM through some process that formats calls to a CLIST as shown in the following:

```
STARTOOL  'SYS1.PROCLIB'  
%CLISTP  'DATA.SET.TO.MANIPULATE'  
%CLISTP  'ANOTHER.DATA.SET'  
%CLISTP  'ONE.MORE'  
. . .
```

Chapter 8: Examples

CLISTP contains a StarTool FDM subcommand to CHANGE to the data set and additional subcommands to manipulate the data set as needed. Any data encountered under CLIST mode (from an implied CLIST call as above or the EXEC subcommand) is assumed to be StarTool FDM subcommands or CLIST commands. If they conflict because their names are the same, use an abbreviated name to reference the StarTool FDM subcommand. For example, to reference IF and CONTROL subcommands, enter the subcommand names, I and CONTRO.

Formatting Data Set VTOC Information

The USAGE subcommand formats the VTOC information for a data set (the VTOC Format 1 and any associated Format 3 records are formatted). This information includes a hexadecimal and character over/under dump of the VTOC entry itself followed by individual fields (as defined in the Debugging Handbooks) formatted with field offset, field name, data value and English description.

To request this information, type an operand on the USAGE subcommand, as shown in the following example:

```
USAGE ALL
```

Formatting Directory Entries

The DIRENTRY subcommand formats directory entries for a member. This information includes a dump format hexadecimal and character display of the directory entry followed by individual fields (as defined in the Debugging Handbooks) formatted with field offset, field name, data value and English description.

Enter DIR (for DIRENTRY) as a MEMLIST line command. Enter DIRENTRY as a primary command for any desired member group. Enter the SHORT operand if you do not want the fields formatted.

Formatting Member Data

The FIND, LIST and REPLACE subcommands format member data identically. FIND and REPLACE display segments of members; that is, only those lines containing a search string.

Seven formats are supported: NUM, NONUM, SNUM, LDUMP, LBLOCK, DUMP and BLOCK. NUM is the default for source data; LDUMP is the default for load members and LBLOCK is the default for VSAM data. When any of these formats is specified for a FIND, LIST or REPLACE subcommand, that format is used for these subcommands until a different format is chosen.

For load members or VSAM data sets, the NUM, SNUM and NONUM formats do not apply (they are equivalent to the default format). Use LDUMP or LBLOCK formats to list or search CSECT data in load members. With these formats, OFFSET(hexoffset) and MODULE(csectname) are supported. For VSAM index or data components, DUMP and BLOCK format physical data records.

Listing Module History

For information on module history, IDR data and COBOL compile options, use the HISTORY subcommand.

If you need translator (compiler or assembler) information by CSECT, type CONTROL TRANSLATOR to ensure that this information is provided by default for the remainder of the current session.

To obtain all history data for a load member, type HISTORY *memname*. If CONTROL NOTRANSLATOR is in effect, StarTool FDM still provides COBOL compile options data.

To obtain translator (and COBOL compile option) data only, type HISTORY *memname translator*.

To obtain USERDATA (or SYSMOD) information only, type HISTORY *memname userdata*.

To obtain ZAP information only, type HISTORY *memname zap*.

You can also search a data set for TRANSLATOR, USERDATA, ZAP or LKED (linkage-editor) information. For example, to search a data set for SYSMOD UZ23546, type HISTORY : *userdata(uz23546)*.

Select members containing specific IDR data by adding a MEMLIST, NEWML or SUBLIST operand to the HISTORY subcommand.

Locating Data in Dialog Tables

Use the LOCATE (or L) command to position to data of interest in a sorted StarTool FDM table (this is not supported in the log, LISTA/DDNAME or WORKPAD). LOCATE searches for data in the field on which the table is sorted.

For example, in a source MEMLIST, the LOCATE command positions to a member name. After the command, SORT ID, LOCATE searches the userid data field.

Managing Data Set Lists

StarTool FDM allows you to create an active data set list similar to ISPF's DSLIST. Unlike DSLIST, however, StarTool FDM allows you to save these lists across sessions, maintain multiple data set lists and add to or exclude data set names from a list. This facility enables you to maintain customized data set lists. To build an initial data set list, type the LISTC primary command.

This command displays a prompting panel that allows you to specify a catalog look-up for various data set qualifiers in a single invocation. Try this using your own TSO userid as the high level qualifier to test this facility. After pressing Enter the LISTC processor displays all matching data sets in a scrollable table. Many commands are available to operate on these data sets. Enter an O line command next to a data set name to get a list of all line commands.

To save this data set list, type *SAVE INTEL* as a primary command. This table is saved as member *##INTEL* in your ISPPROF (profile) data set; you can retrieve it in a later session by typing

LISTC INTEL as a primary command.

Additional invocations of LISTC from an active LISTC table merge new entries in sorted order in the current list. Use the X line command to drop data sets from the list. After obtaining the data sets in the LISTC table, save the list for use in future sessions.

If you want to search disk volumes for data set names, use the LISTF function instead. The input required is similar to the LISTC function except that a volume name or a volume name mask must also be specified.

The LISTC/LISTF functions use a combined table; thus, you can have data sets from a volume source as well as a catalog source. The following line commands are used in LISTC/LISTF applications:

- **=** Repeats the last line command on a different data set
- **C** Changes to a data set
- **DEL** Deletes a data set
- **IDC** Displays the result of an IDCAMS LISTC ENTRY ALL for a data set
- **INFO** Displays VSAM or non-VSAM data set statistics and space use for a data set
- **ML** Changes to a data set and builds a MEMLIST of all members
- **REN** Renames a data set
- **U** Changes to a data set and displays USAGE information
- **X** Drops a data set from the table

Some very powerful primary commands make the LISTC/LISTF function a file management facility. From the primary command line, enter FIND or REPLACE to activate a global function that searches or updates all partitioned data sets in the table.

Managing Member Lists

To obtain a member list containing all members in a data set similar to an ISPF member list, type ALL as a primary command. An equivalent command is ML : since the colon specifies no start or end range and all members are included. If this were a load library, the processing to build the initial member list display would take slightly longer than a source member list because each member is read to obtain the linkage-edit date.

To refer to all members from the current member list with a subcommand, use = as the member group specification. This makes the current member group equal to the members in the member list before performing the operation.

You can save and recall MEMLIST tables. Use the SAVE and MERGE commands to manage saved member list tables. MEMLIST table names can contain one to six alphanumeric characters.

Specify member ranges by using a starting member name followed by a : and then an ending member name. To limit the member list displayed, a selected range can be used as follows: ML ABC:BCD RESET. This example requests a member list to be built starting with members named ABC and ending with members named BCD. The RESET keyword indicates that the member list is to be discarded and rebuilt with the specified member group; otherwise, MEMLIST is an additive process.

Chapter 8: Examples

It is also possible to build a member list of members whose names contain a common string. For example, `ML /ST/` builds a member list containing all members with the character string `ST` somewhere in their name.

Further combinations are possible that limit the member list to members whose names begin with a common string such as `IEB`. In this case, the command is `ML IEB*`.

Member names are not the only potential method for obtaining member lists. The criteria varies with over 80 selectable options from linkage-edit date and module attributes to SSI and IDR data contained within the members. This flexibility expands the ease in identifying the members you want to manipulate. See *Appendix A of the Serena™ StarTool® FDM Reference Guide* for member group specification rules and the MEMLIST function for additional information on filtering members.

A simple approach to managing members consists of building a member list based on the last update or linkage-edit date of members in a library. For example, you can enter either `ML : WEEK` or `WEEK` to build a list of members modified or added in the last week. you can also use the commands `TODAY` and `MONTH` for simple date filtering. Another approach for source members is the `MINE` command which builds a member list of all members whose ID is equal to your `USERID`.

MEMLIST supports line commands with up to four character names. Type a line command in the CMD column next to a member to be manipulated. Type multiple line commands on the same member list screen together. Below are several commonly used member line commands:

- **B** Browses a member using ISPF services
- **COMP** Compares a member with another member that can be in another data set
- **COPY** Copies a member and, optionally, any of its aliases to another data set
- **CS** Builds an ISPF table of all CSECT and ENTRY symbols in a module
- **DEL** Deletes a member
- **E** Edits a source member using ISPF services
- **MAP** Maps the CSECT structure of a load member
- **PR** Prints a member
- **REN** Renames a member
- **S** Defaults to Edit in a source data set; otherwise, it defaults to Browse for load
- **SUB** Submits a member for background processing
- **X** Drops a member from the table
- **XREF** Displays intra-module references

Manipulating Non-Partitioned Data Sets

StarTool FDM supports non-partitioned data sets (sequential, direct and VSAM) in addition to partitioned (source or load) data sets.

Many StarTool FDM subcommands that process member groups also support sequential data sets. These subcommands include ABE, BROWSE, COPY, DCF, DUP, EDIT, FIND, FSE, LIST, PBROWSE, PRINT, REPLACE, REVIEW, SUBMIT, TSOEDIT, TSOLIST, VERIFY and VPRINT. The documented syntax is unchanged except that no data is entered corresponding to any member name position (several of the above subcommands support VSAM and direct data sets in the same way). Many other subcommands support non-partitioned data sets without changes to the documented syntax since no member name parameter is defined for the subcommand.

Examples:

- A LIST subcommand for a partitioned data set looks like:
LIST *memgroup* NONUM

Chapter 8: Examples

- For a non-partitioned data set, a similar subcommand looks like:

```
LIST NONUM
```

For VSAM data sets, the BROWSE, EDIT, DUP, FIND, LIST, REPLACE and VERIFY subcommands support positioning. Positioning can be based on record keys, RBA addresses or relative record numbers depending on the type of VSAM data set. Use it when only specific records of a VSAM data set are of interest.

Migrating Test Modules to Production

Test modules are transferred to production by copying the current production library to a backup library and copying test modules into the production library with REPLACE.

StarTool FDM can improve this process in several ways. Perform the following steps in a batch process:

- In the batch JCL, allocate TEST, PROD and BACKUP files as follows:

```
//TEST          DD  DSN=test.library,DISP=OLD
//PROD          DD  DSN=production.library,DISP=SHR
//BACKUP        DD  DSN=backup.library,DISP=SHR
```

- Enter the test data set: STARTOOL FILE(TEST)
- Start a SYSOUT data set: CONTROL SYSOUT(A)
- Validate all test members: VERIFY :
- Terminate if any errors: CONDEND
- Form an actual list of members: SUBLIST :
- Document all member attributes: ATTRIB * SHORT
- Change to backup library: CHANGE FILE(BACKUP)
- Delete test members: DELETE *
- Compress backup data set: COMPRESS
- Change to production library: CHANGE FILE(PROD)
- Copy members to backup: COPY * FILE(BACKUP)
- Delete production members: DELETE *
- Compress production data set: COMPRESS
- Change to test library again: CHANGE FILE(TEST)
- Copy to production: COPY * FILE(PROD)
- Reset test data set for reuse: FIXPDS RESET

- Close the SYSOUT data set: CONTROL NOSYSOUT

Modifying AMODE and RMODE

To modify the AMODE or RMODE for an entire module, use the ATTR line command from a MEMLIST table. This displays a prefilled entry assist panel that shows the member's current statistics. Change individual attribute entries as desired.

To modify the AMODE or RMODE of a CSECT within a load member, use the MAP command with the MODULE keyword to identify the CSECTs that are to be updated and AMODE24/AMODE31/AMODEANY or RMODE24 /RMODEANY operands to indicate how the addressing or residence modes are to be changed. Use the MAP line command in the CSECT function to update the AMODE or RMODE of a CSECT.

Modifying DCB Parameters

Certain types of user errors can change the DCB recorded for data sets. Change DCB parameters selectively or change all parameters at once as in the command:

```
FIXPDS RECFM(FBA) LRECL(133) BLKSIZE(13300)
```

or for the menu system, type:

```
M.FIXPDS option: Set DCB RECFM    ==> FBA
                  and Set DCB LRECL ==> 133
                  and Set DCB BLKSIZE ==> 13300
```

Note This process only modifies the Format 1 DSCB. Physical blocks in the data set are not affected.

After changing DCB parameters, validate the data set by entering a subcommand such as VERIFY :. This checks for members that are incompatible with the modified DCB parameters. The error members are those added in the original error. If you add the MEMLIST keyword, members with validity errors are added to the MEMLIST table.

Modifying Member Attributes

The recommended way to modify a load member's attributes is with the ATTR line command from a MEMLIST table. This displays a prefilled entry assist panel that reflects the member's current attributes. Change individual attribute entries as desired.

Modifying Member Statistics

To modify a source member's ISPF statistics is with the ATTR line command from a MEMLIST table. This displays a prefilled entry assist panel that reflects the member's current statistics. You can change individual attribute entries as desired. If a member has no ISPF statistics, they can be added.

Modifying Several Members' Attributes

To modify the attributes of several members in a data set, use the ATTRIB subcommand with a member group operand. First, isolate the group of members to be updated.

As an example, assume you need to change the attributes of most RMODEANY members in a data set to RMODE24. Enter a subcommand like MEMLIST : RMODEANY or for the menu system:

```
M.MEMLIST option: Member or member group ==> :  
                and RMODE (residence mode) ==> ANY
```

to build a MEMLIST table with the desired members. Then, drop individual members with an X line command. Finally, change the remaining members with a subcommand like ATTRIB = RMODE24 or for the menu system:

```
M.ATTRIB option: Member or member group ==> =  
                and RMODE (residence mode) ==> 24
```

Obtaining Assistance:

There are many forms of help available in StarTool FDM.

- For an index to tutorials on all StarTool FDM topics, type INDEX
- For a tutorial on the current subcommand or function, press HELP
- For a list of topics managed in a CUA action bar, tab or position the cursor over the area and press ENTER
- For a functional description of a CUA action bar menu item, position the cursor next to the item and press HELP
- For a functional description of any other area in a table panel, position the cursor over the area and press HELP
- For a syntax assist panel for a item in a CUA action bar, position the cursor next to the item and press ENTER

- For an explanation just after receiving warning (PDSnnnW) or error (PDSnnnE) messages, type ?
- For an explanation of any StarTool FDM message (PDSnnn) in the log, position the cursor over the message identifier (the PDSnnn) and press RCHANGE (this is PF key 6 or 18).
- For a directory of available commands, type O (for OPTIONS) as a line command or primary command. This provides a syntax assist panel for the command chosen. As a short-cut, chain options; for example, O.LI requests a syntax assist panel for the LIST subcommand.
- For a menu of available subcommands and functions, type M (for MENU) as a line command or primary command. This provides an entry assist panel for the command. As a short-cut, chain options; for example, M.LI requests an entry assist panel for the LIST subcommand.
- From either a syntax assist or entry assist panel, for a subcommand tutorial, press HELP.

Processing Multiple Data Sets

Use Global commands to process multiple data sets in a LISTA/DDNAME, LISTC/LISTF or WORKPAD table. The GLOBAL command allows any StarTool FDM subcommand to be used on a global basis (against each data set in a table) and the other commands are understood to have a global scope. The following commands are global in LISTA/DDNAME, LISTC/LISTF and WORKPAD tables.

- **COMPRESS** See *“Compressing Data Sets” on page 91* for more information.
- **GLOBAL** See *“Releasing Space in a Data Set” on page 109* for more information.
- **FIND** See *“Searching Multiple Data Sets” on page 113* for more information.
- **MODEL** Changes to each data set in the table and performs a MODEL command to create a data set based on the data set in the table. Use MODEL as a front end to the CREATE subcommand or the IDCAMS subcommand for VSAM data sets.
- **REPLACE** See Updating Multiple Data Sets *“Updating Multiple Data Sets” on page 119* on page Updating Multiple Data Sets for more information.
- **SEEK** Changes to each data set in the table and checks for a member.
- **WHASHO** Performs an ENQUEUE check on all data sets in the table.

Reblocking Source Members

If members in a source data set are blocked too high (for example, from messages produced by the VERIFY subcommand), the REPRO subcommand can reblock these members. You can also specify the maximum physical blocksize to be used with the MAXBLK(blksize) operand; however, it is usually omitted since it defaults to the data set BLKSIZE.

Use the following sequence of subcommands to correct all blocksize errors in a source data set:

1. VERIFY : SUBLIST (to select any members with errors)
2. REPRO * (to reblock any selected members)

Reconstructing Linkage-Edit JCL

Use the MAP subcommand with the JCL (or RELINK) operand if you need to reconstruct linkage-edit JCL and controls for a load member and its aliases. The generated data includes the actual JCL for the linkage-editor with linkage-edit attributes specified in the PARM keyword and linkage-editor control statements to relink a member as it currently exists.

The control statements contain ORDER statements so that CSECTS remain in the same physical order as well as RMODE, AMODE and ALIAS statements. You can edit this data to include data from other sources.

Another use for this data is to isolate a given CSECT into a stand-alone module. To do this, change all ORDER statements into REPLACE statements (use a CHANGE ALL in edit) and remove the names of any CSECTS that are to remain in the final module. Also, change the NAME and any ALIAS statements so that the original members are not overlaid.

Releasing Space in a Data Set

When unused space in a data set needs to be returned to the system, you have several options.

- M.FIXPDS option: Release all free space ====> YES
- FIXPDS RELEASE (to release all free space)
- M.FIXPDS option: Release any free extents ====> YES
- FIXPDS RELEXT (to release all unused extents)
- M.FIXPDS option: Percent of data set free ====> 30
- **FIXPDS RELFREE(30)** (to release up to 30% free space)

Releasing Space in Multiple Data Sets

To release space in multiple data sets, use the GLOBAL command. To reduce prompting, type CONTROL NOPROMPT. Then, build a LISTA/DDNAME, LISTC/LISTF or WORKPAD table containing the data sets to be modified. Finally, from the command line in that table, type GLOBAL FIXPDS option where option is one of the FIXPDS release keywords (RELEASE, RELEXT, RELFREE or RELSAVE).

Renaming Multiple Members

The RENAME subcommand renames multiple members if they have a common prefix. You are prompted with a list of affected members before any actual renames are performed.

As an example, RENAME ABC DEFX GROUP renames member ABCDE to DEFXDE.

Similarly, RENAME ABC ZB GROUP renames member ABCDE to ZBDE.

If the RENAME subcommand changes the name of a main member in a load library, it also updates the directory entries of alias members to reference the renamed main member correctly.

Repairing a CSECT in Multiple Members

To locate and update a CSECT in all members for an entire library, use the following combination of subcommands.

- IF : MODULE(*modname*) THEN(SUBLIST) (selects the correct members)
- MAP * JCL (prepares JCL for relinking these members)
- EDITLOG (enters an edit session on the generated JCL)

Now, add a JOB card, delete member separator messages. Add linkage-edit INCLUDE statements for each member to link in a new version of *modname* and submit the modified JCL.

Restoring Deleted Members

If members are deleted or updated by an editor, the previous version of these members remain in the data set until the data set is compressed. Use the RESTORE subcommand with a MEMLIST option to add the deleted members into a MEMLIST table. Then you can examine these members with several StarTool FDM line commands: DELI, DISA, HIST, LIST, FIND, MAP and REV. After determining which members you want restored, use the REST line command to convert these deleted members back into normal members.

Use either of the following to add deleted members into the MEMLIST:

- M.RESTORE option: Partial member name ==> @@@
- RESTORE @@@ MEMLIST

To reduce the number of deleted members added to the MEMLIST, use the FIND('any string') and the MODULE(modname) operands whenever possible to filter the deleted members that are considered for restoration.

You can recover deleted members even after a data set has been compressed. For this to be possible, however, the deleted members must not have been overwritten by the compress operation itself. Potential members for restoration with this technique include any data recorded after the current DS1LSTAR pointer. This includes data that may not have been initialized in this data set and may contain residual values (garbage).

First, type a FIXPDS MAXSPACE subcommand to reset the DS1LSTAR marker to the end of the allocated space in the data set. Then, restore any deleted members as shown above. Finally, compress the data set again.

Saving a LISTC Table

To build a LISTC/LISTF table, type LISTC to get data sets from a catalog source or type LISTF to get data sets from a volume (VTOC) source. A prompt panel asks you to specify one or more data set name masks. The LISTF prompt panel also requires a volume name or a volume name mask. Once you enter the LISTC/LISTF function, add it to the table by entering LISTC and/or LISTF to search for additional data sets.

Trim the table with the X line command. One very effective way to trim the LISTC/LISTF table is to first sort the table on some criteria (such as SORT TYPE to sort by DSORG) and position to any desired point in the table (by scrolling or the F command) and entering a primary command such as X ABOVE.

Name the table with a command like ID B27 for table B27 and type SAVE to save the table in your profile data set. In a later session, retrieve this table by typing LISTC Update data set statistics by typing the REFRESH command.

Saving a MEMLIST Table

You can save and restore MEMLIST tables. You may want to use saved MEMLIST tables to checkpoint work in progress (perhaps to save where you are working at the end of a day) or to provide a list of members for another application (such as a batch process performing VERIFY : MEMLIST commands to select members with errors and saving the list for interactive review).

To save a MEMLIST table for later reference, type a command like:

```
SAVE mydata
```

Names of saved MEMLIST tables contain one to six alphanumeric characters.

Chapter 8: Examples

To recall a previously saved table, enter a command like `MERGE mydata`.

Note This command adds (or merges) members into any existing MEMLIST table. Add a RESET operand on the command to nullify the current MEMLIST table before adding the saved members.

Searching a Data Set

The FIND subcommand finds members that have a search string and take a conditional action. For example, if all members with the string `//STEPLIB` are to be copied to a different data set and deleted you can do the following:

1. `FIND '//steplib' display then(memlist)` (selects the proper members)
2. `COPY * new.data.set new` (copies the selected members into a clone of the current data set)
3. `DELETE *` (deletes the selected members)

Note the use of the display operand on the FIND subcommand above. Add it to a FIND subcommand if FIND results are displayed in the log in addition to taking a conditional action (such as `then(memlist)`). DISPLAY is particularly useful if you are selecting members for inclusion in a MEMLIST or SUBLIST.

Searching for Panel, Message, or CLIST Members

To find members in concatenated data sets, use the global SEEK command from the DDNAME function. First, build a DDNAME table containing the proper data sets.

For example, type `DDNAME SYSPROC` to search for a CLIST member. Use a DDNAME of `ISPPLIB` for panel members and `ISPMLIB` for message members. If `LIBDEF` or `ALTLIB` data sets are in use, those data sets are not added to the DDNAME table and those data sets are not searched. From the command line in the DDNAME table, type `SEEK memname` where *memname* is the member to be found.

For the menu system, type:

M.10.6 option: Name of members to locate ==> *memname*
and Mask of DDNAMES to search ==> SYSPROC

Searching for System Modules

To find operating system load members and check for redundant system modules, use the FINDMOD subcommand. FINDMOD always performs a system BLDL function to determine if a module is in your STEPLIB or ISPLLIB data sets; however, these data sets are never actually searched individually like the system data sets. Use any of the following:

- M.10.1 (to search by name)
- M.10.2 (to search by address)
- FINDMOD *modname* SYSTEM/ADDRESS

You can request that StarTool FDM change to the data set by adding CHANGE or add GO to establish a new GO session. With either of these options, add NUMBER(*n*) to specify which library to activate.

Searching Multiple Data Sets

To find data in multiple data sets, use the global FIND command. First, build a LISTA/DDNAME, LISTC/LISTF or WORKPAD table containing the data sets to be searched. All data sets should be partitioned or non-partitioned.

In a LISTC/LISTF table, the global FIND command bypasses non-partitioned data sets. If you want to search non-partitioned data sets, type a command similar to the following:

```
GLOBAL FIND 'string' ...
```

Finally, from the command line in that table, type a FIND command to go to the global FIND syntax assist panel. If all of the data sets are partitioned, add a member group specification such as : to search all members. Otherwise, omit this positional parameter.

Selecting Members

The FIND subcommand filters members by contents and the IF subcommand filters members by attributes. For these subcommands, the first operand is a member group (see “*Specifying a Member Group*” on page 115).

Chapter 8: Examples

For the IF subcommand, enter attributes next (for example, enter RENT, AMODE24 or AUTH for load members). All specified conditions must be met to select a member; these are *and* conditions. Finally, specify an action to take if true (with THEN(*command*)) and an action to take if false (with ELSE(*command*)). Normally, specify THEN(SUBLIST) so that following subcommands can be specified with operands on the members selected. For example, use

```
IF copy* ID(myid) LAST(7) THEN(sublist)
PRINT * SYSOUT(a) DEST(rmt21)
COPY * other.data.set
```

to print selected members updated recently by *myid* and copy them to another data set

For the FIND subcommand, enter the character string being searched for after the member group specification. It is specified as a delimited string with 1 to 32 characters. The first character is the final delimiter. The delimiter must not be used within the string itself. In practice, a single quote (') or double quote (") is used as FIND string delimiters. If you use X as the string delimiter, the data is assumed to be hexadecimal digits and the string may contain 2 to 64 hexadecimal digits.

Following the string specification, other parameters can be in any order. You can specify FIND boundary conditions (WORD, SUFFIX or PREFIX), FIND limits (MAXIN, MAXOUT, MAXFIND, MODULE, OFFSET, ...) and FIND list format (NUM, NONUM, SNUM, LDUMP, LBLOCK, DUMP or BLOCK).

To select members with a given string and certain conditions, specify an action to take if found (with THEN(*command*)) and an action to take if not found (with ELSE(*command*)). You can specify FIND limits (MAXIN, MAXOUT, MAXFIND, MODULE, OFFSET, ...) and FIND list format (NUM, NONUM, SNUM, LDUMP, LBLOCK, DUMP or BLOCK). Specify THEN(SUBLIST) so that following subcommands can be specified with operands on the members selected. For example, use

```
FIND COPY* 'PGM=IEBCOPY' DISPLAY THEN(SUBLIST)
PRINT * SYSOUT(A) DEST(RMT21)
DELETE *
```

to print JCL members that invoke IEBCOPY and delete them.

Specifying a Member Group

All subcommands that allow a member group specification (except for DISPLAY and PATTERN) also modify the current member group. These subcommands specify lists of members and member groups. If you specify a list of members, it must be enclosed in parentheses and individual items in the list must be separated with one or more blanks or a comma.

Specify individual member group items using the following entry types:

Range	Example: MEMBERS start:end AB:C includes all members beginning with AB through all members beginning with C. C: includes all members beginning with C through the end of the members. F: includes all members up through members beginning with F. : includes all members.
Pattern	Example: MEMBERS part1/part2 AB/C includes all members whose names contain AB anywhere and C anywhere. ABC/ includes all members containing ABC anywhere. /ABC includes all members containing ABC anywhere. You can use placeholders (% or ?) that match any single character anywhere in a pattern specification.
Combination	Example: MEMBERS start*end AB*C includes all members whose names start with AB and contain C at the end of the member name. ABC* includes all members whose names start with ABC. *ABC includes all members containing ABC at the end of the member name. You can use placeholders (% or ?) that match any single character anywhere in a combination specification.

Chapter 8: Examples

Placeholder	Example: MEMBERS start%end A%?C includes all members with four-character names beginning with A and ending with E. A% includes all members with two-character names beginning with A. ?F includes all members with two-character names ending with F. You can use placeholders (% or ?) that match any single character anywhere in a pattern or combination specification.
Asterisk	Example: MEMBER * Use a single asterisk to refer to the current member group.
Equal	Example: MEMBERS = Use an equal symbol to indicate that the current member group is to be set equal to the list of members in the current MEMLIST before passing control to the subcommand.

The current member group is maintained across CHANGE subcommands until a subcommand specifies a different member group. An example that includes several of the elements above is:

```
MEMBERS (AB:C, ABC/FX?Z, CO%Y*, A%?E)
```

A MEMBERS subcommand, such as the above, causes the associated members in the data set to be listed. To convert this to a sublist (a list of member names), follow this with a subcommand like:

```
SUBLIST * ALIAS
```

(where ALIAS indicates that associated members also be included).

Switching Modes

Each of the major ISPMODE functions is maintained in parallel for the duration of the StarTool FDM session. An exception is made for the MEMLIST, CSECTS and ZAP functions. These functions terminate automatically at a CHANGE subcommand since they only apply to members in a specific data set.

To explicitly go from one function to another, enter mode switch commands. In general, enter the name of the function that you want to continue (for pending, uncompleted line commands) or switch to (at the position last maintained in that function). Thus, ML is a mode switch command for MEMLIST. LA or DD is a mode switch command for a LISTA/DDNAME table established earlier.

If you enter a function name with operands, MEMLIST and LISTV add to an existing table while CSECTS, DDNAME and LISTA rebuild their tables. Other functions, such as LOG and GO, establish a new table or session or place you back in a previously established table or session.

If a function has uncompleted (or pending) line commands, you can select them automatically in a dynamic hierarchy by the END command. Enter a STATUS command to see the current state of all functions in logical order. A function can have INACTIVE (never activated or not currently active due to a CHANGE subcommand), ACTIVE (currently active) or PENDING ACTION (one or more uncompleted line commands) status. An END command selects PENDING functions starting from the first function listed in the STATUS panel.

If you enter an END command and no functions have pending line commands, these actions occur:

- the primary options panel displays (if you selected the standard user interface in SETALL)
- the last menu system panel displays (if you did not enter parallel mode commands since the last use of the menu system)
- a termination prompting panel displays for the current GO session
- or, you are presented with StarTool FDM itself if only a single GO session is active (if termination prompting is active in SETALL)
- or, the current GO session terminates, or StarTool FDM itself terminates (if termination prompting is not active in SETALL).

However, if you activated the standard user interface in SETALL, StarTool FDM always prompts before terminating to verify that you want to exit the StarTool FDM environment.

Trimming Dialog Tables

You can trim all StarTool FDM dialog tables with the X primary command. The supported operands (ABOVE, BELOW and ALL) trim the table in relation to the top line in the current table display. For tables that support line commands, use the X line command or an XX block line command pair to drop individual table lines.

Chapter 8: Examples

REMOVE and EXCLUDE are available for all function tables. From a log or MEMLIST table, use the EXCLUDE subcommand to trim members from the MEMLIST table.

The EXCLUDE subcommand supports the same attribute filtering operands that are available on a MEMLIST subcommand. From any table, use the REMOVE command to trim all table elements that match a search string (or do not match a search string with the NOT keyword). The REMOVE command supports the same operands as are available on the F command with the exception of the directional keywords (NEXT, FIRST and LAST) because the REMOVE command always processes an entire table.

For tables that can be sorted, sort the table in a particular order (for example, with load members, enter SORT DATE (to sort by linkage-edit date), position to some point in the table by scrolling, with the F command or with the LOCATE command and enter an X primary command to trim the entire table. Finally, enter SORT (with no operands) to sort the table back into its default order.

Updating a Data Set

To update multiple members in a data set, use the REPLACE subcommand to update in place. Enter a command similar to the following:

```
REPLACE memgroup 'datanow' 'datanew'
```

or if you want to use delimited strings, enter:

```
REPLACE memgroup /datanow/datanew/
```

For the menu system, enter:

```
M.REPLACE option: Member or member group ==> memgroup
                   Find data ==> datanow
                   Replacement data ==> datanew
                   Update data on disk ==> NO
```

By default, the above subcommand performs only a trial update. After examining the output from the REPLACE subcommand, recall the subcommand and add the WRITE keyword or change the M.REPLACE panel to:

```
Update data on disk ==> YES
```

to update the members.

To enter hexadecimal data in the REPLACE subcommand, use a special format with X delimiters:

```
REPLACE memgroup XhexbeforeXhexafterX
```

Updating Multiple Data Sets

To update data in multiple data sets, use the global REPLACE command. First, build a LISTA/DDNAME, LISTC/LISTF or WORKPAD table containing the data sets to be updated. All data sets must be partitioned or non-partitioned.

In a LISTC/LISTF table, the global REPLACE command bypasses non-partitioned data sets. If you want to update non-partitioned data sets, enter a command similar to:

```
GLOBAL REPLACE 'string1' 'string2' ...
```

From the command line in that table, enter a REPLACE command to go to the global REPLACE syntax assist panel. If all of the data sets are partitioned, add a member group specification such as :. Otherwise, omit this positional parameter. If you do not enter the WRITE keyword, REPLACE only performs a trial update. Repeat the REPLACE command with the WRITE operand to update the data sets.

Validating Data

To validate members of an entire data set, use the VERIFY subcommand. For a PDS, if the member group name is : (a colon), the entire data set is checked and error members are identified. Otherwise, members are checked individually and statistics are generated for each member.

To check individual members, enter VERIFY A:B (where A:B is the group of members to be checked).

To validate an entire PDS, enter VERIFY :

For the menu system, enter:

```
M.VERIFY option: Member or member group ==> A:B
```

Add the MEMLIST option to select members with errors.

By default for VSAM data sets, VERIFY performs an IDCAMS VERIFY operation on the data set followed by an IDCAMS EXAMINE for KSDS data sets. Then, the VERIFY subcommand accesses a KSDS data set by key and in addressed mode to verify that all records are accessible.

Zapping a CSECT

To update a load member interactively, use the ZAP command. First, build a CSECTS table using the member to be modified. For example, if the ABEXIT module is to be updated, enter CSECTS ABEXIT as a primary command. Then, position to the CSECT to be modified by scrolling, or with the LOCATE command or the F command and type ZAP as a line command next to the CSECT of ABEXIT that is to be modified. Next, position within the CSECT by scrolling, or with the F command or with the LOCATE command.

Type over the hexadecimal data or the character data displayed. Press Enter to see the *CHA (change) marker on modified lines.

To commit all changes to disk, type ZAP as a primary command.

To back out all changes not yet written to disk, type UNDO as a primary command.

To display AMASPZAP compatible control statements with no updates, type BUILD as a primary command.

To terminate the ZAP function with no additional changes on disk, type NOZAP as a primary command.

FIXPDS SUBCOMMAND

9

StarTool FDM has several facilities for modifying the current data set. Many data set problems are better addressed if the data set is not open while its attributes are being changed. In StarTool FDM terminology for the FIXPDS subcommand, this is an alternate data set.

ACTIVE DATA SETS

The following examples show a normal FIXPDS subcommand and its results when performed on active data sets.

Sample FIXPDS for the Active Data Set

```
----- ISPMODE Session# 1 Log# 1 ----- Row 77 to 81 of 81
COMMAND ==> fix                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSDSCB/ -----
>----->c lib.cntlr
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C  FB           80    1600   1X    10         9    10 TRK    8

PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSDSCB/
*****Bottomofdata*****
```

Sample FIXPDS to Add 10 Disk Tracks

```
----- FIXPDS: Modify a Partitioned Data Set -----
OPTION ==>

- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSMSCB/ -----

Change the directory:
Add directory blocks ==> (1 to 9999) used: 2
Percent free blocks ==> (1 to 999) Total tracks: 10
Release disk space: free: 9
Release all free space ==> NO (YES or NO)
Release any free extents ==> NO (YES or NO)
Percent of data set free ==> (1 to 999)

Add a disk extent:
Add disk tracks ==> 10 (1 to 99999)
Add disk cylinders ==> (1 to 99999)
Percent of data set free ==> (1 to 999)

Modify data set attributes:
Management class ==> (migration and backup attributes)
Storage class ==> (storage unit, volume, storage service)
Expiration date ==> (Julian -- yyyyddd)
Secondary space type ==> TRK (BLK, CYL, TRK or RDNBLK)
Secondary amount ==> 10 (0 to 99999 in space type units)

Modify DCB parameters:
Set DCB LRECL ==> 80 (1 to 32767)
Set DCB BLKSIZE ==> 1600 (1 to 32767)
Set DCB RECFM ==> FB (F/FB/FA/FM/FBA/FBM/V/VB/VBA/VBM/U)
Set DCB OPTCD ==> C (W, C, Z, WC, WZ, CZ, WCZ or NO)

Modify other parameters:
Reset directory entries ==> (1 to 9999; use * for the same number)
Check for EDIT sessions ==> NO (YES or NO for use with Reset)
```

Sample FIXPDS Prompt

```

----- Reply Required ----- Enter a prompt response
COMMAND ==> yes                                SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSMSCB/ -----
>----->Fixpds ADDTRK(10)

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C  FB           80    1600   1X    10         9    10 TRK    8

PDS392A Should this data set be modified (Yes/No) ?
*****Bottomofdata*****
    
```

Sample FIXPDS Result

```

----- ISPMODE Session# 1 Log# 1 ----- Row 82 to 92 of 92
COMMAND ==>                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSMSCB/ -----
>----->Fixpds ADDTRK(10)

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C  FB           80    1600   1X    10         9    10 TRK    8

PDS392A Should this data set be modified (Yes/No) ?
>----->yes

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C  FB           80    1600   2X    20        19    10 TRK    8

***** Bottom of data *****
    
```

Changing the *FORMAT1 DSCB*

FIXPDS also supports changing fields in the *FORMAT1 DSCB* directly if you set your Interface level to Advanced or Power in the SETALL option.

Chapter 9: FIXPDS Subcommand

Sample SETALL Panel to Change Interface Level

```
----- Set Combined Defaults -----  
OPTION ==>  
To save these variables in your ISPF profile as defaults for future StarTool  
sessions, enter SAVE as a primary command or press PF6 or PF18  
More:      +  
  
Set Environmental Options:  
Interface level      ==> advanced (Standard/Advanced/Power)  
Automatic EDITLOG   ==> YES (Yes/No/Lin; Lin affects line commands onl  
Termination prompt  ==> YES (Yes/No/Bac; Bac requests backup processin  
CUA mode action bars ==> YES (Yes/No)  
Global prompt       ==> YES (Yes/No)  
Internal Jump       ==> YES (Yes/No)  
Separator character ==> ; (use a special character)
```

Sample FIXPDS Subcommand

```
----- ISPMODE Session# 1 Log# 1   Interface Level Updated  
COMMAND ==> fix                                SCROLL ==> CSR  
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSDSDB/ -----  
  
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR  
PDS200I SHR  3380 C  FB      80   1600   2X   20      19   10 TRK      8  
  
*****Bottomofdata*****
```

Sample FIXPDS to Update Current Data Set

```

OPTION ==>

- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSMSCB/ -----
                                                    More:  +
Change the directory:                               Total directories: 10
  Add directory blocks ==> (1 to 9999)                used: 2
  Percent free blocks ==> (1 to 999)                 Total tracks: 20
Release disk space:                                free: 19
  Release all free space ==> NO (YES or NO)
  Release any free extents ==> NO (YES or NO)
  Percent of data set free ==> (1 to 999)
Add a disk extent:
  Add disk tracks ==> 10 (1 to 99999)
  Add disk cylinders ==> (1 to 99999)
  Percent of data set free ==> (1 to 999)
Modify data set attributes:
  Management class ==> (migration and backup attributes)
  Storage class ==> (storage unit, volume, storage service)
  Expiration date ==> (Julian -- yyyyddd)
  Secondary space type ==> TRK (BLK, CYL, TRK or RNDBLK)
  Secondary amount ==> 10 (0 to 99999 in space type units)
Modify DCB parameters:
  Set DCB LRECL ==> 80 (1 to 32767)
  Set DCB BLKSIZE ==> 1600 (1 to 32767)
  Set DCB RECFM ==> FB (F/FB/FA/FM/FBA/FBM/V/VB/VBA/VBM/U)
  Set DCB DSORG ==> PO (PS/PSU/DA/DAU/PO/POU/IS/ISU/VS/VSU)
  Set DCB OPTCD ==> C (W, C, Z, WC, WZ, CZ, WCZ or NO)
Modify other parameters:
  Reset directory entries ==> (1 to 9999; use * for the same number)
  Check for EDIT sessions ==> NO (YES or NO for use with Reset)
  Correct member order ==> NO (YES or NO)
  Set logical end-of-file ==> (hexadecimal TTR or MAX for all space)
Update the Format-1 DSCB:
  Current data set ==> yes (YES or NO)
  Alternate data set name ==>
  Alternate volume ==> (required if not cataloged)

```

Sample FORMAT1 DSCB Prompt

```

----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Current - DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDSDCB/ -----
DSCB1 at 000A000017

More: +

Caution: many of the fields of the Format 1 DSCB can not be changed without
compromising data set integrity. For more information using field
level help, place the cursor on any of these items and press HELP

OFF,LEN LABEL HEX VALUE DESCRIPTION
(0,44) DS1DSNAM ==> WSER07.LIB.CNTLR
(2C,1) DS1FMTID ==> F1 Format identifier
(2D,6) DS1DSSN ==> E2C5D9F0F0F1 Data set serial name
(33,2) DS1VOLSQ ==> 0001 Volume sequence number
(35,3) DS1CREDIT ==> 610056 Creation date
(38,3) DS1EXPDT ==> 000000 Expiration date
(3B,1) DS1NOEPV ==> 02 Number of extents on volume
(3C,1) DS1NOBDB ==> 00 Number of bytes used in last directory
(3D,1) ==> 00 Reserved
(3E,6) DS1SYS CD ==> C9C2D4D6E2E5 System code
(44,7) ==> E2F24040404040 System code (last 7 characters)
(4B,3) DS1REFD ==> 610073 Date last referenced
(4E,1) DS1SMSFG ==> 00 System managed storage indicators
(4F,1) DS1SCXTF ==> 00 Secondary space extension flag
(50,2) DS1SCXTV ==> 0000 Secondary space extension value
(52,2) DS1DSORG ==> 0200 Data set organization
(54,1) DS1RECFM ==> 90 Record format
(55,1) DS1OPTCD ==> 20 Option code
(56,2) DS1BLKL ==> 0640 Block length
(58,2) DS1LRECL ==> 0050 Logical record length
(5A,1) DS1KEYL ==> 00 Key length
(5A,1) DS1KEYL ==> 00 Key length
(5B,2) DS1RKP ==> 0000 Relative key position
(5D,1) DS1DSIND ==> 82 Data set indicator flags
(5E,4) DS1SCALO ==> 8000000A Secondary allocation type and amount
(62,3) DS1LSTAR ==> 00001C TTR of last used track and block of data
(65,2) DS1TRBAL ==> 3F80 Bytes remaining on last track used
(67,2) ==> 0000 Reserved
(69,10) DS1EXT1 ==> 0100007A0000007A0009 Extent 1 in XX00CCCCHHHHCCCCHHHH
(73,10) DS1EXT2 ==> 010100AC0000A00AD0004 Extent 2 in XX01CCCCHHHHCCCCHHHH
(7D,10) DS1EXT3 ==> 000000000000000000000000 Extent 3 in XX02CCCCHHHHCCCCHHHH
(87,6) DS1PTRDS ==> 0000000000 CCHHR of any associated Format 3 DSCB

```

If you place the cursor over any field and press HELP or PF1, field level help displays for the item. See the following panel for the DS1SCALO field.

Sample FORMAT1 DSCB Field Level Help for DS1SCALO

```

----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Current - DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDSCB/ -----
DSCB1 at 000A000017

More: -

(3D,1)      == +----- DS1SCALO -----+
(3E,6) DS1SYSCD == | TUTORIAL - Format 1 DSCB: DS1SCALO field - TUTORIAL |
(44,7)      == | OPTION => |
(4B,3) DS1REFD == | OFFSET: 5E to 61 |
(4E,1) DS1SMSFG == | The DS1SCALO field contains Secondary Allocation |
(4F,1) DS1SCXTF == | Space parameters. This field is used to determine |
(50,2) DS1SCXTV == | how to allocate a secondary extent for a data set. |
(52,2) DS1DSORG == | |
(54,1) DS1RECFM == | The first byte contains bit flags and the next three |
(55,1) DS1OPTCD == | characters contains the related amount in binary. |
(56,2) DS1BLKL == | |
(58,2) DS1LRECL == | The bit flags have the following meanings: |
(5A,1) DS1KEYL == | DS1CYL or X'C0' Request space in cylinders |
(5B,2) DS1RKP == | DS1TRK or X'80' Request space in tracks |
(5D,1) DS1DSIND == | DS1AVR or X'40' Request space by average block |
(5E,4) DS1SCALO == | DS1AVRND or X'41' Average block and ROUND up |
(62,3) DS1LSTAR == | DS1EXT or X'10' Use DS1SCEXT extension instead |
(65,2) DS1TRBAL == | DS1CONTG or X'08' Request contiguous space |
(67,2)      == | DS1DSABS or X'00' Request space by absolute track |
(69,10) DS1EXT1 == | |
(73,10) DS1EXT2 == | Be careful in manipulating these values to modify |
(7D,10) DS1EXT3 == | the secondary allocation for a data set. |
(87,6) DS1PTRDS == | |
+-----+

```

Sample FORMAT1 DSCB with DS1SCALO Updated

```
----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Current - DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDS DSCB/ -----
DSCB1 at 000A000017

(5B,2) DS1RKP ==> 0000 Relative key position
(5D,1) DS1DSIND ==> 82 Data set indicator flags
(5E,4) DS1SCALO ==> 80000030 Secondary allocation type and amount
(62,3) DS1LSTAR ==> 00001C TTR of last used track and block of data
(65,2) DS1TRBAL ==> 3F80 Bytes remaining on last track used
(67,2) ==> 0000 Reserved
(69,10) DS1EXT1 ==> 0100007A0000007A0009 Extent 1 in XX00CCCCHHHCCCCCHHH
(73,10) DS1EXT2 ==> 010100AC000A00AD0004 Extent 2 in XX01CCCCHHHCCCCCHHH
(7D,10) DS1EXT3 ==> 00000000000000000000 Extent 3 in XX02CCCCHHHCCCCCHHH
(87,6) DS1PTRDS ==> 0000000000 CCHHR of any associated Format 3 DSCB

More: -
```

Sample FIXPDS Prompt for a FORMAT1 DSCB Change

```
----- Reply Required ----- Enter a prompt response
COMMAND ==> yes SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDS DSCB/ -----
>----->Fixpds DSCB(5E 8000000A 80000030)

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 1600 2X 20 19 10 TRK 8

PDS392A Should this data set be modified (Yes/No) ?
*****Bottomofdata*****
```

Sample FIXPDS DSCB Update Results

```

----- ISPMODE Session# 1 Log# 1 ----- Row 93 to 103 of 103
COMMAND ==>
                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=PDS DSCB/ -----
>----->Fixpds DSCB(5E 8000000A 80000030)

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 1600 2X 20 19 10 TRK 8

PDS392A Should this data set be modified (Yes/No) ?
>----->y

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 1600 2X 20 19 48 TRK 8

*****Bottomofdata*****
    
```

FIXPDS of an Alternate Data Set

To modify a data set other than the current active data set, you must set your Interface level to Advanced or Power in the SETALL option as shown in the previous example.

Sample FIXPDS to Modify an Alternate Data Set

```

----- ISPMODE Session# 1 Log# 1 Interface Level Updated
COMMAND ==> fix
                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM= -----

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 11440 1X 1000 718 200 TRK 68

PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTLR,VOL=SER=SER001 MEM=
*****Bottomofdata*****
    
```

Sample FIXPDS Prompt Specifying an Alternate Data Set

```

----- FIXPDS: Modify a Partitioned Data Set -----
OPTION ==>

- DSN=WSER07.LIB.CNTL,VOL=SER=SER001 MEM= -----
                                                    More:  +
Change the directory:                               Total directories:  88
  Add directory blocks    ==>          (1 to 9999)          used:    20
  Percent free blocks     ==>          (1 to 999)           Total tracks: 1000
Release disk space:
  Release all free space  ==> NO          (YES or NO)
  Release any free extents ==> NO        (YES or NO)
  Percent of data set free ==>          (1 to 999)           free:    718
Add a disk extent:
  Add disk tracks         ==>          (1 to 99999)
  Add disk cylinders      ==>          (1 to 99999)
  Percent of data set free ==>          (1 to 999)
Modify data set attributes:
  Management class       ==>          (migration and backup attributes)
  Storage class          ==>          (storage unit, volume, storage service)
  Expiration date        ==>          (Julian -- yyyyddd)
  Secondary space type   ==> TRK        (BLK, CYL, TRK or RNDBLK)
  Secondary amount       ==> 200       (0 to 99999 in space type units)
Modify DCB parameters:
  Set DCB LRECL          ==> 80        (1 to 32767)
  Set DCB BLKSIZE        ==> 11440    (1 to 32767)
  Set DCB RECFM          ==> FB        (F/FB/FA/FM/FBA/FBM/V/VB/VBA/VBM/U)
  Set DCB DSORG          ==> PO        (PS/PSU/DA/DAU/PO/POU/IS/ISU/VS/VSU)
  Set DCB OPTCD          ==> C         (W, C, Z, WC, WZ, CZ, WCZ or NO)
Modify other parameters:
  Reset directory entries ==>          (1 to 9999; use * for the same number)
  Check for EDIT sessions ==> NO      (YES or NO for use with Reset)
  Correct member order   ==> NO      (YES or NO)
  Set logical end-of-file ==>          (hexadecimal TTR or MAX for all space)
Update the Format-1 DSCB:
  Current data set       ==> NO        (YES or NO)
  Alternate data set name ==> lib.cntls
  Alternate volume       ==>          (required if not cataloged)

```

Sample FIXPDS Alternate Data Set Prompt

```

----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Note: this is not the current data set
  Data set: WSER07.LIB.CNTLS
  Volume:   SER001
  DSCB1 at: 000A000018

More: +

Caution: many of the fields of the Format 1 DSCB can not be changed without
compromising data set integrity.  For more information using field
level help, place the cursor on any of these items and press HELP

OFF,LEN LABEL      HEX VALUE      DESCRIPTION
(0,44) DS1DSNAM ==> WSER07.LIB.CNTLS
(2C,1) DS1FMTID ==> F1          Format identifier
(2D,6) DS1DSSN ==> E2C5D9F0F0F1 Data set serial name
(33,2) DS1VOLSQ ==> 0001       Volume sequence number
(35,3) DS1CREDT ==> 610056     Creation date
(38,3) DS1EXPDT ==> 000000     Expiration date
(3B,1) DS1NOEPV ==> 01         Number of extents on volume
(3C,1) DS1NOBDB ==> 00         Number of bytes used in last directory
(3D,1)          ==> 00         Reserved
(3E,6) DS1SYSCD ==> C9C2D4D6E2E5 System code
(44,7)          ==> E2F24040404040 System code (last 7 characters)
(4B,3) DS1REFD  ==> 610073     Date last referenced
(4E,1) DS1SMSFG ==> 00         System managed storage indicators
(4F,1) DS1SCXTF ==> 00         Secondary space extension flag
(50,2) DS1SCXTV ==> 0000       Secondary space extension value
(52,2) DS1DSORG ==> 0200       Data set organization
(54,1) DS1RECFM ==> 90         Record format
(55,1) DS1OPTCD ==> 20         Option code
(56,2) DS1BLKL ==> 7FD0       Block length
(58,2) DS1LRECL ==> 0050      Logical record length
(5A,1) DS1KEYL  ==> 00         Key length
(5B,2) DS1RKP   ==> 0000       Relative key position
(5D,1) DS1DSIND ==> 82         Data set indicator flags
(5E,4) DS1SCALO ==> 8000000A    Secondary allocation type and amount
(62,3) DS1LSTAR ==> 000206     TTR of last used track and block of data
(65,2) DS1TRBAL ==> A300       Bytes remaining on last track used
(67,2)          ==> 0000       Reserved
(69,10) DS1EXT1 ==> 0100007A000A007B0004 Extent 1 in XX00CCCCHHHHCCCCHHHH
(73,10) DS1EXT2 ==> 00000000000000000000 Extent 2 in XX01CCCCHHHHCCCCHHHH
(7D,10) DS1EXT3 ==> 00000000000000000000 Extent 3 in XX02CCCCHHHHCCCCHHHH
(87,6) DS1PTRDS ==> 0000000000 CCHHR of any associated Format 3 DSCB
  
```

Chapter 9: FIXPDS Subcommand

Sample FIXPDS Alternate Data Set DS1SCALO Modification

The following change to DS1SCALO changes the secondary allocation type to cylinder and the amount to 1.

```
----- FIXPDS: Format 1 DSCB Modification -----  
OPTION ==>  
  
Note: this is not the current data set  
Data set: WSER07.LIB.CNTLS  
Volume: SER001  
DSCB1 at: 000A000018  
  
More: -  
  
(5B,2) DS1RKP ==> 0000 Relative key position  
(5D,1) DS1DSIND ==> 82 Data set indicator flags  
(5E,4) DS1SCALO ==> c0000001 Secondary allocation type and amount  
(62,3) DS1LSTAR ==> 000206 TTR of last used track and block of data  
(65,2) DS1TRBAL ==> A300 Bytes remaining on last track used  
(67,2) ==> 0000 Reserved  
(69,10) DS1EXT1 ==> 0100007A000A007B0004 Extent 1 in XX00CCCCHHHHCCCCHHHH  
(73,10) DS1EXT2 ==> 00000000000000000000 Extent 2 in XX01CCCCHHHHCCCCHHHH  
(7D,10) DS1EXT3 ==> 00000000000000000000 Extent 3 in XX02CCCCHHHHCCCCHHHH  
(87,6) DS1PTRDS ==> 0000000000 CCHHR of any associated Format 3 DSCB
```

Sample FIXPDS Alternate Data Set Update Prompt

```
----- Reply Required ----- Enter a prompt response  
COMMAND ==> yes SCROLL ==> CSR  
Reply required -- Enter YES or NO  
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001 MEM=PDS DSCB/ -----  
>----->Fixpds DSCB(5E 8000000A C0000001) MODDSNAME('WSER07.LIB.CNTLS') MODVOL  
-----UME(SER001)  
PDS398A Should this alternate data set be modified (Yes/No) ?  
*****Bottomofdata*****
```

Sample FIXPDS Reply

```

----- ISPMODE Session# 1 Log# 1 ----- Row 68 to 71 of 71
COMMAND ==> c lib.cntls                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSDSCB/ -----
>----->Fixpds DSCB(5E 8000000A C0000001) MODDSNAME('WSER07.LIB.CNTLS') MODVOL
-----UME(SER001)
PDS398A Should this alternate data set be modified (Yes/No) ?
>----->yes
*****Bottomofdata*****
    
```

Sample FIXPDS Results

```

----- ISPMODE Session# 1 Log# 1 ----- Row 72 to 76 of 76
COMMAND ==>
- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSDSCB/ -----
>----->c lib.cntls
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C  FB           80  32720  1X   10       7    1 CYL      6

PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSDSCB/
*****Bottomofdata*****
    
```

Restoring a Deleted Data Set

StarTool FDM can restore deleted data sets with either of two options: REST or RJCL. REST performs all of its functions online but causes volume space problems in the disk freespace pool. You can correct this space problem by turning on the volume DIRF bit. Use the RJCL method since it does not cause volume freespace problems.

Sample VMAP with RJCL to RESTORE a DATA SET

```
----- VMAP WORK01 ----- Row 1 to 7 of 7
COMMAND ==>                               SCROLL ==> CSR
Enter an ISPF command or a special control code:
-----
CMD  CCHH      End CCHH EX# DATA/MSG -----DATA SET NAME -----
00000001 0008000E  1 *---134 *** FREE SPACE EXTENT *** _____
00090000 0009000E  1 *---15 SYS1.VTOCIX.WORK01_____
000A0000 000B000E  1 *---30 *** FORMAT 4 (VTOC) EXTENT *** _____
000C0000 004E0009  1 *--1000 WSER07.LIB.CNTILO_____
rjcl 004E000A 005B000E  1 *---200 *** FREE SPACE EXTENT *** _____
005C0000 0065000E  1 *---150 WSER07.LIB.CNTLS_____
00660000 0374000E  1 *-11745 *** FREE SPACE EXTENT *** _____
***** Bottom of data *****
```

Sample RJCL Prompt for Data Set Name and JOBCARD

```
----- ABSTR allocation BATCHJCL job card prompt -----
COMMAND ==>

Data set name to be allocated:
DSN ==> lib.restore

Specify Job cards for the batch job to be created to allocate
on VOLUME: WORK01 at location CCHH: 004E000A (relative track: 1180 )
for length in tracks: 200

==> //WSER07A JOB (X170,374), 'SHOWS RESTORE',CLASS=A,
==> //  MSGCLASS=X,TIME=(0,30),NOTIFY=WSER07
==> //
==> //
```

Note: This uses absolute allocation which is not supported on SMS managed volumes. The resulting data set must have its DSORG and other attributes corrected using FIXPDS specifying this data set in the "Alternate data set name" field.

Sample RJCL edit session with a SUB Command to Restore the Data Set

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
WSER07.SPFTEMP1.CNTL                               Columns 00001 00072
Command ==> sub                                     Scroll ==> CSR
***** ***** Top of Data *****
000001 //WSER07A JOB (X170,374), 'SHOWS RESTORE', CLASS=A,
000002 //  MSGCLASS=X, TIME=(0,30), NOTIFY=WSER07
000003 //AA EXEC PGM=IEFB14
000004 //SYSLIB DD DSN=WSER07.LIB.RESTORE, DISP=(NEW,CATLG), VOL=SER=WORK01,
000005 // UNIT=SYSALLDA, SPACE=(ABSTR, (200,1180))
***** ***** Bottom of Data *****
    
```

Sample VMAP after Return From Edit and JOB Executed

```

----- VMAP WORK01 ----- Row 1 to 7 of 7
COMMAND ==> c lib.restore                               SCROLL ==> CSR
Enter an ISPF command or a special control code:
-----
CMD  CCHH      End CCHH EX# DATA/MSG -----DATA SET NAME -----
00000001 0008000E 1 *---134 *** FREE SPACE EXTENT ***
00090000 0009000E 1 *---15 SYS1.VIOCIX.WORK01
000A0000 000B000E 1 *---30 *** FORMAT 4 (VIOC) EXTENT ***
000C0000 004E0009 1 *--1000 WSER07.LIB.CNTLO
004E000A 005B000E 1 *RJCL *** FREE SPACE EXTENT ***
005C0000 0065000E 1 *---150 WSER07.LIB.CNTLS
00660000 0374000E 1 *--11745 *** FREE SPACE EXTENT ***
***** ***** Bottom of data *****
    
```

Modify the data set DCB parameters and move the DS1LSTAR pointer to the end.

Chapter 9: FIXPDS Subcommand

Sample First Panel for LIB.RESTORE

```
----- ISPMODE Session# 1 Log# 1 --- Row 237 to 243 of 243
COMMAND ==> fix recfm(fb) lrecl(80) blksize(32720) maxspace SCROLL ==> CSR
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 -----
>----->c lib.restore
PDS200I DISP UNIT      RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY DSORG
PDS200I SHR  3380      ?        0         0   1X   200     200     0 TRK **

PDS530W This data set is not partitioned

PDS300A ENTER OPTION -- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01
***** Bottom of data *****
```

Sample FIXPDS Reply

```
----- Reply Required ----- Enter a prompt response
COMMAND ==> yes SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 -----
>----->fix recfm(fb) lrecl(80) blksize(32720) maxspace
PDS580W DCB changes only affect the data set attributes

PDS200I DISP UNIT      RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY DSORG
PDS200I SHR  3380      U         0         0   1X   200     200     0 TRK **

PDS392A Should this data set be modified (Yes/No) ?
***** Bottom of data *****
```

Sample LIST Subcommand to Examine Data Set Contents

```

----- ISPMODE Session# 1 Log# 1 --- Row 244 to 255 of 255
COMMAND ==> list dump maxlen(32) maxout(300)                SCROLL ==> CSR
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 -----
>----->fix recfm(fb) lrecl(80) blksize(32720) maxspace
PDS580W DCB changes only affect the data set attributes

PDS200I DISP UNIT      RECFM LRECL BLKSIZE   ALLOCTRK FREETRK  SECONDARY DSORG
PDS200I SHR  3380      U          0      0      1X   200    200    0 TRK  **

PDS392A Should this data set be modified (Yes/No) ?
>----->yes

PDS200I DISP UNIT      RECFM LRECL BLKSIZE   ALLOCTRK FREETRK  SECONDARY DSORG
PDS200I SHR  3380      FB          80  32720  1X   200      0    0 TRK  PS

***** Bottom of data *****

```

Since this is a PDS data set (note the 256 byte blocks below), change the DSORG to match the data set.

Sample FIXPDS to Change the Data Set to Partitioned

```

----- ISPMODE Session# 1 Log# 1 --- Row 268 to 294 of 309
COMMAND ==> fixpds dsorg(po)                                SCROLL ==> CSR
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 -----
>----->list dump maxlen(32) maxout(300)
PDS140I DUMP RECORD          1  LENGTH   256  TTR 000001
  000000 0000 00FEE2E3 C1D9C2C1 E3E30000 130F0101 *..      .....*
  000010 0010 00020097 087F0097 087F1432 0011000E *...p."p.".....*

PDS140I DUMP RECORD          2  LENGTH   256  TTR 000002
  000100 0000 000EFFFF FFFFFFFF FFFF0000 00000000 *.....*
  000110 0010 00000000 00000000 00000000 00000000 *.....*

...
PDS142I 10 blocks in this data set
***** Bottom of data *****

```

Chapter 9: FIXPDS Subcommand

Sample FIXPDS Reply

```
----- Reply Required ----- Enter a prompt response
COMMAND ==> yes                                SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 -----
>----->fix dsorg(po)

PDS200I DISP UNIT      RECFM LRECL BLKSIZE  ALLOCTRK FREETRK  SECONDARY DSORG
PDS200I SHR  3380      FB          80  32720   1X   200      0    0 TRK PS

PDS392A Should this data set be modified (Yes/No) ?
***** Bottom of data *****
```

Sample FIXPDS Reply with D to Display Members

```
----- ISPMODE Session# 1 Log# 1 --- Row 310 to 320 of 320
COMMAND ==> d                                SCROLL ==> CSR
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01 MEM=: -----
>----->fix dsorg(po)

PDS200I DISP UNIT      RECFM LRECL BLKSIZE  ALLOCTRK FREETRK  SECONDARY DSORG
PDS200I SHR  3380      FB          80  32720   1X   200      0    0 TRK PS

PDS392A Should this data set be modified (Yes/No) ?
>----->yes

PDS200I DISP UNIT      RECFM LRECL BLKSIZE  ALLOCTRK FREETRK  SECONDARY FREEDIR
PDS200I SHR  3380      FB          80  32720   1X   200      0    0 TRK      8

***** Bottom of data *****
```

The data set is restored successfully. COMPRESS the data set to make the available space in the data set available for new PDS members.

Sample DISPLAY Output Showing Member Names

```

----- ISPMODE Session# 1 Log# 1 --- Row 561 to 562 of 562
COMMAND ==> compress                                SCROLL ==> CSR
- DSN=WSER07.LIB.RESTORE,VOL=SER=WORK01  MEM=: -----
>----->d
***** Bottom of data *****
    
```

FIXPDS to Delete or Rename a Data Set

FIXPDS deletes or renames the current data set or an alternate data set even if it is ENQUEUED. Set your Interface level to Advanced or Power in the SETALL option.

To guard against deleting an active system data set, StarTool FDM requires that the data set be uncataloged before you rename or delete it. If the data set name assigned in the FIXPDS FORMAT1 DSCB panel begins with NULLR, the data set is renamed and deleted. If the data set is renamed to any other string beginning with NULL, the data set is deleted (indexed volumes are supported). Otherwise, the data set is just renamed. Any method that renames a data set does not support data sets on an indexed volume.

Sample LISTC Table With UNC to Uncatalog the Data Set

```

----- List files 0      - (Attributes) ----- Row 1 to 7 of 7
COMMAND ==>
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=PDSE  -----
CMD  C V DATA/MSG -----DATA SET NAME ----- VOLUME DO RECFM LRECL BLKSI
   Y Y          WSER07.LIB.CNTL_____ SER001 PO FB      80 11440
   Y Y          WSER07.LIB.CNTLO_____ WORK01 PO FB      80 11440
   Y M          WSER07.LIB.CNTLPE_____ OS39H2 PE FB      80 11440
   Y Y          WSER07.LIB.CNTLR_____ SER001 PO FB      80 1600
unc Y Y          WSER07.LIB.CNTLS_____ SER001 PO FB      80 32720
   Y Y          WSER07.LIB.CNTLT_____ SER001 PO FB      80 32720
   Y Y          WSER07.LIB.CNTL82_____ SER001 PO FB      82 23452
*****Bottomofdata*****
    
```

Sample LISTC Table with FIX to Invoke FIXPDS

```
----- List files 0      - (Attributes) ----- Row 1 to 7 of 7
COMMAND ==>                                SCROLL ==> CSR
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002 MEM=PDSE -----
CMD  C V DATA/MSG -----DATA SET NAME ----- VOLUME DO RECFM LRECL BLKSI
     Y Y          WSER07.LIB.CNTL_____ SER001 PO FB      80 11440
     Y Y          WSER07.LIB.CNTLO_____ WORK01 PO FB      80 11440
     Y M          WSER07.LIB.CNTLPE_____ OS39H2 PE FB      80 11440
     Y Y          WSER07.LIB.CNTLR_____ SER001 PO FB      80 1600
fix  N Y *UNCAT*  WSER07.LIB.CNTLS_____ SER001 PO FB      80 32720
     Y Y          WSER07.LIB.CNTLT_____ SER001 PO FB      80 32720
     Y Y          WSER07.LIB.CNTL82_____ SER001 PO FB      82 23452
***** Bottom of data *****
```

Sample Partial FIXPDS Prompt to Update Current Data Set

```
----- FIXPDS: Modify a Partitioned Data Set -----
OPTION ==>

- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSE -----
...
Modify other parameters:
  Reset directory entries ==>          (1 to 9999; use * for the same number)
  Check for EDIT sessions ==> NO      (YES or NO for use with Reset)
  Correct member order   ==> NO      (YES or NO)
  Set logical end-of-file ==>        (hexadecimal TTR or MAX for all space)
Update the Format-1 DSCB:
  Current data set       ==> yes      (YES or NO)
  Alternate data set name ==>
  Alternate volume       ==>          (required if not cataloged)
```

Sample FIXPDS FORMAT1 DSCB Modification Prompt

```

----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Current - DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSE -----
DSCB1 at 000A000018

More:      +

Caution: many of the fields of the Format 1 DSCB can not be changed without
compromising data set integrity. For more information using field
level help, place the cursor on any of these items and press HELP

OFF,LEN LABEL      HEX VALUE      DESCRIPTION
(0,44) DS1DSNAM ==> WSER07.LIB.CNTLS
(2C,1) DS1FMTID ==> F1              Format identifier
(2D,6) DS1DSSN ==> E2C5D9F0F0F1     Data set serial name
    
```

Sample FIXPDS FORMAT1 DSCB DS1DSNAM Update

In the following panel, the data set name is typed over to delete the data set.

```

----- FIXPDS: Format 1 DSCB Modification -----
OPTION ==>

Current - DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSE -----
DSCB1 at 000A000018

More:      +

Caution: many of the fields of the Format 1 DSCB can not be changed without
compromising data set integrity. For more information using field
level help, place the cursor on any of these items and press HELP

OFF,LEN LABEL      HEX VALUE      DESCRIPTION
(0,44) DS1DSNAM ==> NULLX
(2C,1) DS1FMTID ==> F1              Format identifier
(2D,6) DS1DSSN ==> E2C5D9F0F0F1     Data set serial name
    
```

Chapter 9: FIXPDS Subcommand

Sample FIXPDS Update Prompt

```
----- Reply Required ----- Enter a prompt response
COMMAND ==> yes                               SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSE -----
>----->Fixpds NEWDSNAME('NULLX')
PDS291I 'WSER07.LIB.CNTLS' is not allocated

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 32720 1X 10 7 1 CYL 6

PDS399A Should this data set be renamed (Yes/No) ?
```

StarTool FDM changed to a different data set because the active data set was deleted.

Sample FIXPDS Results

```
----- ISPMODE Session# 1 Log# 1 --- Row 147 to 163 of 163
COMMAND ==>                               SCROLL ==> CSR
- DSN=SYS1.PROCLIB,VOL=SER=SCPMV5 MEM=PDSE -----
>----->Fixpds NEWDSNAME('NULLX')
PDS291I 'WSER07.LIB.CNTLS' is not allocated

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 C FB 80 32720 1X 10 7 1 CYL 6

PDS399A Should this data set be renamed (Yes/No) ?
>----->yes

PDS200I DISP UNIT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR 3380 FB 80 32720 1X 30 23 1 CYL 14

***** Bottom of data *****
```

IBM BATCH UTILITIES COMPARISON

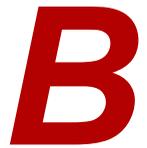


IBM Utility	Function	StarTool FDM	Description
AMBLIST	List load member attributes	ATTRIB	List all attributes of a module including the linkage-edit date
"	"	CSECTS	Display CSECTS in a selectable table
"	"	HISTORY	List IDR data (zap, userdata and translator information)
"	"	MAP	Map internal CSECT structure or rebuild linkage-edit statements
"	"	MEMLIST	Display members in a selectable table
"	"	XREF	Cross-reference CSECT and EXTERNAL symbols
IEBCOPY	Copy members	COPY	Copy members to an existing or new data set. By default, include all member aliases. COPYMOD reblocking of load members is requested, if necessary. COPY also copies data to a sequential data set.
"	"	DUP	Copy members without calling another program. Convert records from FB to VB. Convert records from VB to FB. Convert CLIST members.
"	Compress PDS data sets	COMPRESS	Compresses data sets and checks if the data set is in the linklist. Checks if the data set is managed by LLA.
IEBGENER	Copy sequential data sets	DUP	Copy members to sequential data sets or copy sequential data sets.
IEHLIST	VTOC list utility	LISTF	Data sets in a selectable table with statistics. Delete or rename data sets. Catalog or uncatalog data sets.
"	"	LISTV	List attributes and available space on disk volumes.

Chapter A: IBM Batch Utilities Comparison

IBM Utility	Function	StarTool FDM	Description
"	"	USAGE	Map the Format 1 DSCB in the log (with the ALL operand).
IEWL	Linkage-edit	ATTRIB	Alter most load member linkage attributes.
"	"	ALIAS	Define a member alias.
IEBUPDTE	Unload/reload members	COMBINE	Combine members using IEBUPDTE control statements preserving ISPF statistics.
"	"	SEPARATE	Separate combined members into original members.
IDCAMS	IDCAMS Utilities	IDCAMS	Format IDCAMS commands using a menu approach with results in the log.
"	"	LISTC	List data sets from the MVS catalog in a selectable, sortable table.
"	"	IDC	LISTC line command to display an IDCAMS LISTCAT with ENTRY and ALL.
"	"	INFO	LISTC line command to display VSAM space use and attributes.
"	"	DUP	Similar to IDCAMS REPRO (but only sequential or PDS member output) with <i>shared</i> access.
"	"	VERIFY	Perform IDCAMS VERIFY and EXAMINE, also validate data.
AMASPZAP	Zap load members	ZAP	Zap load members with character or hexadecimal overtyping. Update with zap IDR history or provide AMASPZAP controls.

ISPF UTILITIES COMPARISON



StarTool FDM offers functions equivalent to most of the ISPF functions through its menu system. For example, if you want to use the equivalent of ISPF 3.4 in StarTool FDM, enter M.3.4 or 3.4. If you want to perform an ISPF function and return to StarTool FDM, stack an ISPF session with a command like ISPF 3.4.

ISPF	Function	StarTool FDM	Description
=1	Browse	BROWSE	Browse a member, data set or VSAM file.
=2	Edit	EDIT	Edit a member, data set or VSAM file (SAVE is not supported for VSAM).
=3.1	Library Applications	ALIAS	Define a member alias.
"	"	ATTRIB	List or alter member statistics or module attributes.
"	"	BROWSE	Browse members.
"	"	DELETE	Delete members.
"	"	RENAME	Rename members.
"	"	VERIFY	Validate members.
=3.2	Data Set Applications	CREATE	Create a new data set.
"	"	DSNAME	Display summary data set information.
"	"	MODEL	Create a new data set with overtyping (including VSAM).
"	"	USAGE	Display additional data set information.
"	"	USAGE ALL	Display detailed data set information.
=3.3	Data Set Move/Copy	COPY	Copy selected members; by default, this includes all aliases unlike =3.3 COPYMOD is requested if load member reblocking is required.
"	"	DUP	Copy selected members with possible conversion from FB to VB or vice-versa including CLIST conversion.

Chapter B: ISPF Utilities Comparison

ISPF	Function	StarTool FDM	Description
"	"	REPRO	Copy members to a new name in the same data set.
=3.4	Data Set List	PLIST	List a saved data set table from anywhere in ISPF.
"	"	LISTC	List data sets using a catalog source.
"	"	LISTF	List data sets using a volume source.
"	"	MASK	Add selected data sets to a LISTC/LISTF table.
"	"	LISTA	List data sets allocated to a TSO session by attributes.
"	"	DDNAME	List data sets allocated to a TSO session by DDNAME mask.
"	"	WORKPAD	Display selected data set names and commands in a selectable table.
=3.5	Reset Stats	ATTRIB	List or alter member statistics or module attributes.
=3.8	Command Table	CMDTBL	Display and modify an ISPF command tables.
=3.13	Compare Members	COMPARE	Compare two members using SUPERC, COMPAREX or COMPARE\$.
"	"	COMPDIR	Compare multiple members for similarities or differences.
=3.14	Search for	FIND	Search members and perform conditional processing.
"	"	REPLACE	Update strings in members.

GLOSSARY

Alias member

A member whose directory entry has a value that indicates that the member entry is an alternate name for a main member; a main member and its aliases are associated with the same recorded data in a data set.

Apparent alias member

Data in a PDS that is pointed to by more than one main directory entry.

Associated members

Members with the same TTR address. For a main member, the associated members could be alias members or an apparent alias member; for an alias member, the associated members could be other aliases or a main member.

Batch mode

A mode of StarTool FDM operation. In batch mode, all user communication is performed with PUTGET, GETLINE and PUTLINE TSO services. ISPF services are available. In this mode of operation, StarTool FDM operates under control of the batch Terminal Monitor Program (the TMP, or IKJEFT01).

BLDL

A BPAM function used to check for the existence of members.

BPAM

Basic Partitioned Access Method. BPAM is similar to BSAM that is for sequential data. StarTool FDM uses BPAM for member check operations and always uses BPAM for directory updates and reading PDSE data sets.

BSAM

Basic Sequential Access Method. StarTool FDM uses BSAM for several utility functions.

CCHHR

Cylinder, Cylinder, Head, Head, Record which is a ten-byte hexadecimal disk address relative to the start of the volume.

Command

A request for an operation. For example, STARTOOL is a command and FIXPDS is a StarTool FDM subcommand. The distinction between a command and a subcommand is not important.

Compress

A process that removes deleted members from a PDS. Disk space formerly occupied by these deleted members is made available for new members. In a PDSE data set, space from deleted members is managed dynamically. You cannot compress a PDSE.

Deleted member

Member data that is not pointed to by a directory entry. Deleted members remain in a PDS until the data set is compressed. In a PDSE data set, space from deleted members is managed dynamically. You cannot restore deleted members.

Directory

A segment of a PDS that is a sequential data set with eight-byte keys and 256-byte data records containing pointers to data in the PDS member portion of the data set. For a PDSE, PDS directory and member data is created in a virtual PDS data set.

Directory entry

A logical entry in a PDS or PDSE directory that points to its associated data in the member data. It consists of an eight-character member name field and one or more three byte relative address (TTR) fields.

DS1LSTAR

A pointer in the data set control block (DSCB) that indicates the last used disk address for a data set. This pointer is updated after a member is added to a PDS data set or after a PDS is compressed.

EXCP

EXecute Channel Program. This access method is used extensively by StarTool FDM to read an entire disk track with a single operation.

Function

A request for an ISPF (or ISPMODE) operation. Most ISPMODE functions support their own commands.

ISPMODE

A mode of StarTool FDM operation. With ISPMODE, StarTool FDM operates as an ISPF dialog and normal ISPF services (HELP, TSO, SPLIT, SWAP,...). All StarTool FDM services are available.

Line command

A command entered on a table line in the CMD field. Line commands may be up to four characters long.

Line mode

A mode of StarTool FDM operation. In line mode, all user communication is performed with PUTGET, GETLINE and PUTLINE TSO services. No ISPF services are available.

Load library

A PDS or PDSE that has record format U and is used for executable modules.

Main member

A non-alias directory entry and its associated recorded data.

Member

Data in a PDS or PDSE that is pointed to by one or more directory entries. Members may be alias or main members. Deleted members that have no associated directory entries. For a PDSE, space from deleted members is managed dynamically. You cannot restore deleted members or compress a PDSE.

Module

A member in a load library. Also known as a load module or load member.

Orphan member

A member marked as an alias that does not have an associated main member.

PDS or Partitioned data set

PDS. A data set with fixed, variable or undefined format that is used as a *library* for related types of information. Partitioned data sets contain two segments of data: a directory of information and member data. This type of library is requested as DSNTYPE(PDS) in JCL or a TSO ALLOCATE.

PDSE or Partitioned Data Set Extended

PDSE. An SMS-managed data set that is logically similar to a PDS. This type of library is requested as DSNTYPE(LIBRARY) in JCL or on an ALLOCATE command. A PDSE can be accessed by BPAM or BSAM access methods to obtain a virtual data set that looks like a PDS.

Primary command

A subcommand or command entered from the command line of a panel.

QSAM

Queued Sequential Access Method. StarTool FDM uses QSAM for several utility functions.

Source library

A PDS or PDSE data set that has fixed or variable format data. Source libraries are used to contain non-executable data.

StarTool FDM

The premier programmer's workbench for MVS environments. A multipurpose ISPF dialog and TSO command processor that manipulates data sets and members in groups or individually.

Subcommand

A request for an operation that is within the scope of work requested by the previously issued command. STARTOOL is a command. FIXPDS is a StarTool FDM subcommand. The distinction between a subcommand and a command is not important.

TTR

Track, Track, Record (a 1- to 6-digit hexadecimal disk address relative to the start of the data set). This type of address is stored in the member directory entry to indicate the start of a member.

Symbols

= 41
? 106

A

ADDCYL 85
ADDDIR 85
adding
 a single extent in cylinders 81
 a single extent in tracks 81
 an entry point 81
 cylinders, single extent 85
 directory blocks 81, 85
 entry point 85
 tracks, single extent 85
 ZAP
 IDR records 81, 86
ADDTRK 85
advanced options
 described 39
Advanced Options menu 38
advanced users 25
ALIAS
 operand 91
 subcommand 85
ALIASCHK command 89
allocating
 a data set 81, 86
AMODE
 modifying 105

assistance
 obtaining 83
ATTR line command 105
ATTRIB
 subcommand 85, 105, 106

B

background
 executing in 97
 JCL 66, 69, 71
backing up
 multiple members 81, 86
batch JCL 66, 69, 71
batch mode 31, 65, 97
building
 a member list 81
 See creating

C

Calling StarTool
 Batch 66
 Dialog 47
 Dynamic TSO 68
 From a program 68
 ISPF 32, 47
 ISPMODE in Batch 69
 Line mode 29
 Started Task 32
chained line command 36
CHANGE operand 91
changing
 secondary allocations 82

Index

- CHECK keyword 95
- checking
 - alias members 82
- checkpointing
 - the log 82
- CLIST members
 - converting 93
 - searching 112
- CLIST mode 97
- CLISTs
 - in batch 49
 - in ISPMODE 48
 - interactively 49
 - invoking with CMD command 95
 - processing guidelines 56
- cloning a member 82
- CMD 95
- COMBINE subcommand 90
- combining multiple members 82
- command field 48
- command line 27
- command processor mode 31
- commands
 - ALIASCHK 89
 - chained 24
 - DCONTROL 50, 51
 - DGETVAR 50, 55
 - DSETVAR 50, 53
 - dynamic 75
 - EDITLOG 24, 110
 - EDITTBL 25
 - EXCLUDE 72, 96, 117
 - F 72
 - FIND 25, 72
 - function 75
 - GLOBAL 108
 - HELP 106
 - INDEX 27, 106
 - ISPF 75
 - jump 41
 - L 73, 99
 - line 20, 75
 - LOCATE 73, 99
 - MODEL 86
 - OUTPUT 25, 73
 - PANEL 50, 56
 - primary 75
 - RCHANGE 25
 - RFIND 25
 - SEEK 112
 - SETALL 20
 - SETUSER 94
 - SORT 72
 - STANDARD 20
 - SUSPEND 29
 - UT 94
 - X 73, 96, 117
- COMP line command 91
- COMPARE subcommand 90
- comparing
 - members 82
 - multiple members 82

COMPDIR subcommand 91

COMPRESS

command 92

global command 108

subcommand 91

compressing

data sets 82

multiple data sets 82

computer resources

using 77

concepts 75

CONDEND subcommand 97

confirmation panel 16

controlling

GO sessions 82

multiple logs 82

converting

CLIST members 82, 93

source members 82, 93

COPY

command 94

subcommand 94

copying

members 82

creating

member list 87

CSECTS

function 120

switching 116

zapping 120

cylinders

adding single extent 85

D

data

validating 119

data set

allocating 86

data set lists 20, 100

data set repair 123, 129, 133, 139

data set space

releasing 109

data sets

compressing 91

compressing multiple 92

current 32

direct 103

multiple, processing 108

restoring 133

searching 112

sequential 103

updating 118

updating multiple 119

VSAM 103

data/message field 48

DCB parameters

modifying 105

DCONTROL

command 50

command operands 51

DDNAME

function 112

defining

a dynamic command 82, 94

DELETE

deleting multiple members 95

subcommand 95

deleted members

restoring 110

deleting

all members 82

multiple members 82

DGETVAR

command 50

command operands 55

dialog processing 47

Index

dialog tables
 locating 99
 trimming 117
direct data sets 103
directory blocks
 adding 85
directory entries
 formatting 98
DIRENTRY
 subcommand 98
DISASM
 subcommand 95
disassembling members 82, 95
displaying
 external references 82, 96
DSETVAR
 command 50, 53, 54
 operands 53
DUP subcommand 93

E

Edit macro
 ADDCYL 85
 ADDDIR 85
 ADDTRK 85
EDIT subcommand 96
editing
 multiple members 82, 96
EDITLOG
 command 24, 57, 110
EDITTBL 25
entry assist panels 23
entry point
 adding 85

EXCLUDE
 command 72, 96, 117
 copying members 94
 facilities 72
 subcommand 89, 96, 117
excluding multiple members 82, 96
executing 97
 batch mode 97
 in background 82
 in CLIST mode 82
EXIST 94
express mode 52
external references
 displaying 96

F

F command 72
FIND
 command 72
 global command 108, 113
 subcommand 89, 98, 112, 113
FINDMOD subcommand 113
FIXPDS 109, 121, 123, 129, 133, 139
 subcommand 85, 89, 95, 105, 109, 110
formatting
 data set VTOC information 83
 directory entries 83, 98
 member data 83, 98
formatting VTOC information 98

functions

CSECTS *120*
 DDNAME *112*
 ISPMODE *29*
 LC *100, 111*
 LF *100, 111*
 LISTC *100, 111*
 LISTF *100, 111*
 MEMLIST *87, 101, 111*
 ML *87, 101, 111*
 ZAP *120*

G

getting support *14*
 GLOBAL *108*
 command *108, 109*
 data set type filter names *71*
 global command *108, 109*
 global commands
 COMPRESS *108*
 FIND *108, 113*
 GLOBAL *108, 109*
 MODEL *108*
 REPLACE *108, 119*
 SEEK *108*
 WHOHAS *108*
 global processing *70*
 GO
 controlling sessions *92*

H

help
 obtaining *106*
 HELP Command *106*
 hierarchical mode *40*
 HISTORY subcommand *99*

I

I/O
 reducing *78*
 IF subcommand *89, 110, 113*
 IND\$FILE *29*
 INDEX
 primary command *27*
 information
 module history *99*
 See also translator data *99*
 See also USERDATA information *99*
 See also Zap information
 interface levels
 advanced *20*
 power *20*
 standard *20*
 internal jump *41*
 invoking StarTool FDM *32*
 in READY mode *47*
 in the background *47*
 primary options panel *47*
 subcommands *95*
 through another TSO command *47*
 through ISPF *47*
 ISPF panel applications
 invoking with XPANEL *95*
 ISPF program applications
 invoking with PGM command *95*
 ISPF utilities *145*
 ISPMODE *20*
 CLISTs *48*
 dialog operation *32*
 function *29*
 in batch *69*
 subcommand *32*
 using CLISTs in *48*

Index

ISPMODE subcommands
 ISPMODE *33*
 ISPXEQ *33*
 MEMLIST *33*

J

JCL *66, 69, 71*
jump commands *41*

K

key data
 entering *16*

L

L Command *73, 99*
LC Function *100, 111*
LF Function *100, 111*
libraries
 LLA managed *76*
license information *34*
line commands *20, 75*
line mode *29*
linkage-edit JCL, reconstructing *109*
LIST
 subcommand *98*
list member specification *115*
LISTC
 function *100, 111*
 primary command *100*
 saving table *111*
LISTF Function *100, 111*
listing module history *83*
LLA managed libraries *76*
LOCATE command *73, 99*

locating
 data in dialog tables *83*
 dialog tables *99*
log
 checkpointing *90*
logs
 controlling multiple *92*

M

managing
 data set lists *83, 100*
 member lists *83, 101*
manipulating
 data sets *103*
 non-partitioned data sets *83*
MAP subcommand *105, 109, 110*
member attributes
 modifying *105, 106*
member data
 formatting *98*
member group
 specifying *115*
member lists
 creating *87*
 managing *101*
member statistics
 modifying *106*
members
 cloning *90*
 combining multiple *90*
 comparing *90*
 comparing multiple *91*
 copying *94*
 deleted *77*
 deleting all *95*
 deleting multiple *95*
 disassembling *95*
 selecting *113*

- MEMLIST
 - function *87, 101, 111*
 - saving table *111*
 - switching *116*
 - menu system *20*
 - menus
 - advanced *36*
 - Primary Options *35*
 - messages
 - PDS#102M *53*
 - searching *112*
 - migrating
 - test files *104*
 - test modules to production *83*
 - ML Function *87, 101, 111*
 - mode
 - switching *116*
 - mode switch command *40*
 - MODEL
 - command *86*
 - global command *108*
 - modes
 - advanced *25*
 - batch *31, 65, 97*
 - CLIST *97*
 - express *52*
 - hierarchical *40*
 - line *29*
 - parallel *40*
 - power *28*
 - single subcommand *33*
 - standard *20*
 - started task *32*
 - switching *116*
 - modifying
 - AMODE *105*
 - AMODE and RMODE *83*
 - DCB parameters *83, 105*
 - member attributes *83, 105, 106*
 - member statistics *83, 106*
 - RMODE *105*
 - several members' attributes *83*
 - module history *99*
 - multiple data sets
 - updating *119*
 - multiple members
 - backing up *86*
 - editing *96*
 - excluding *96*
 - renaming *110*
 - repairing CSECTs *110*
 - MVS utilities *143*
- ## N
- navigating *40*
 - NOCHANGE operand *91*
 - NOEXIST *94*
 - non-partitioned data sets *103*
- ## O
- obtaining
 - help *106*
 - translator data *99*
 - USERDATA information *99*
 - ZAP information *99*
 - operands
 - DCONTROL *51*
 - DGETVAR *55*
 - DSETVAR *53*
 - PANEL *56*

Index

Option 13
 Examine 41
OPTIONS 30
options
 advanced 39
 current hardware and software environment 43
 customized defaults 45
 Examine Current Environment 41
 for systems programmers 41
OUTPUT 25
 command 73

P

paging 78
 reducing 78
PANEL
 command 50
 command operands 56
 command syntax 56
panels
 Advanced Options 38
 searching 112
parallel mode 33, 40, 48
pattern member specification 115
PDS#102M 53
PDSE 32
PGM 95
placeholder member specification 115
power mode 28
power users 28
primary commands
 commands 75
 defined 75
 dynamic 75
 function 75
 ISPF 75
 subcommands 75

Primary Options panel 21, 35, 47
processing multiple data sets 83, 108
production
 migrating files to 104
programmers
 power mode 28
publications 12
pull down menus 27
PUTLINE interface 29

R

range
 member specification 115
RCHANGE 25
reblocking source members 83, 108
RECEIVE 29
reconstructing linkage-edit JCL 83, 109
releasing
 data set space 109
 space in a data set 83
 space in multiple data sets 83
RENAME subcommand 110
renaming multiple members 83, 110
repairing
 CSECTs in multiple members 83, 110
REPLACE 94
 global command 108, 119
 subcommand 98, 118
replacing existing files
 EXIST command 94
replacing existing members
 REPLACE command 94
REPRO 108
 subcommand 86, 108
REPRO line command 90
REPRO Subcommand 86
REPRO subcommand 86

RESTORE subcommand *110*
 restoring deleted data set *133*
 resurrecting deleted members *83*
 REXX *49*
 RFIND *25*
 RMODE
 modifying *105*

S

saving
 LISTC table *84, 111*
 MEMLIST table *84, 111*
 SDSF *32*
 search conditions
 specifying *16*
 searching
 a data set *84*
 CLIST members *112*
 data sets *112*
 for panel, message or CLIST
 members *84*
 for system modules *84*
 messages *112*
 multiple data sets *84, 113*
 panels *112*
 system modules *113*
 SEEK
 command *112*
 global command *108*
 selecting members *84, 113*
 SEND *29*
 SEPARATE subcommand *90*
 sequential data sets *103*
 SER10TY *13*
 SERENA *9, 26, 34*
 SETALL command *20*
 SETUSER command *94*

short-cut commands *88*
 single subcommand mode *33*
 SORT command *72*
 source members
 converting *93*
 reblocking *108*
 specifying
 member group *84, 115*
 STANDARD command *20*
 standard mode *20*
 start parameters
 defining IMS *16*
 started task mode *32*
 subcommands
 ALIAS *85*
 ATTRIB *85, 105, 106*
 COPY *94*
 DELETE *95*
 DIRENTRY *98*
 DISASM *95*
 DUP *93*
 EDIT *96*
 EXCLUDE *117*
 FIND *98, 112, 113*
 FINDMOD *113*
 FIXPDS *85, 89, 95, 105, 109, 110*
 HISTORY *99*
 IF *110, 113*
 invoking *95*
 LIST *98*
 MAP *105, 109, 110*
 OPTIONS *30*
 RENAME *110*
 REPLACE *98, 118*
 REPRO *86, 108*
 RESTORE *110*
 USAGE *98*
 VERIFY *119*
 XREF *96*

Index

support

Serena *14*

SUSPEND *29, 32*

switching modes *84, 116*

syntax

DGETVAR command *55*

DSETVAR command *53*

PANEL command *56*

syntax assist panels *27*

system modules

searching *113*

user modes

advanced *25*

power *28*

USERDATA information

obtaining *99*

users

advanced *25*

power *28*

UT command *94*

utilities

ISPF *145*

MVS *143*

T

Terminal Monitor Program *31*

test files

migrating to production *104*

TMP *31*

tracks

adding single extent *85*

translator data

obtaining *99*

TRAP *95*

trimming

dialog tables *84*

member lists *89*

tables *117*

TSO commands

invoking with CMD command *95*

invoking with TRAP command *95*

TSO environment *68*

U

updating

a data set *84*

multiple data sets *84, 119*

USAGE subcommand *98*

V

validating data *84, 119*

VERIFY subcommand *119*

VSAM data sets *103*

VTOC

formatting information *98*

reducing contention *79*

W

WHOHAS global command *108*

workbench concept *78*

X

X

command *73, 96, 117*

line command *89*

primary command *89*

XPANEL *95*

XREF subcommand *96*

Z

ZAP

- adding IDR records *86*
- function *120*
- obtaining information *99*
- switching functions *116*
- zapping CSECTs *84, 120*

