



StarTool[®] FDM

System Services

© Copyright 2001-2020 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Product version: 7.7.1

Publication date: March 2020 (rebranded only)

Table of Contents

	Welcome to StarTool FDM	7
	About This Book	7
	Before You Begin	7
	Conventions	7
	Documentation	9
	Related Publications.	9
	Using the PDF Documentation.	10
<i>Chapter 1</i>	Advanced Options Menu	11
	Advanced Options Menu	11
	Examine Environment Settings	12
	Current Hardware and Software Environment	14
	Customized Defaults	15
<i>Chapter 2</i>	CLIST Processing	17
	External and Internal CLISTs	17
	Using CLISTs in ISPMODE.	17
	DCONTROL Command	18
	Syntax	18
	Operands	19
	DSETVAR Command.	20
	Syntax	20
	Message PDS#102M	20
	DGETVAR Command	21
	Syntax	22
	PANEL Command	22
	Syntax	22
	CLIST Processing Guidelines	28
<i>Chapter 3</i>	Library Lookaside Processing	31
	LLA-Managed Libraries	31
	LLA and Deleted Members.	31
<i>Chapter 4</i>	FIXPDS Subcommand	33
	Modifying PDS Attributes for an Active Data Set	33
	Sample FIXPDS for the Active Data Set	33
	Sample FIXPDS Prompt	33
	Sample FIXPDS Result	34
	Sample FIXPDS to Add 10 Disk Tracks	34
	Changing the FORMAT1 DSCB for an Alternate Data Set	35
	SETALL Panel to Change Interface Level	35

FIXPDS Subcommand	35
FIXPDS to Update Current Data Set.	36
FORMAT1 DSCB Prompt	37
Sample FORMAT1 DSCB Field Level Help for DS1SCALO.	38
Sample FORMAT1 DSCB with DS1SCALO Updated.	38
FIXPDS Prompt for a FORMAT1 DSCB Change.	39
Sample FIXPDS DSCB Update Results	39
Sample FIXPDS to Modify an Alternate Data Set	39
FIXPDS Prompt Specifying an Alternate Data Set	40
Sample FIXPDS Alternate Data Set Prompt.	41
FIXPDS Alternate Data Set DS1SCALO Modification	42
Sample FIXPDS Alternate Data Set Update Promp.	42
Sample FIXPDS Reply	42
Sample FIXPDS Results	43
Restoring a Deleted Data Set	43
Sample VMAP with RJCL to RESTORE a DATA SET.	43
Sample RJCL Prompt for Data Set Name and JOBCARD	44
Sample RJCL edit session with a SUB Command to Restore the Data Set	44
VMAP after Return From Edit and JOB Executed	44
Sample First Panel for LIB.RESTORE	45
Sample FIXPDS Reply	45
LIST Subcommand to Examine Data Set Contents.	45
Sample FIXPDS to Change the Data Set to Partitioned.	46
Sample FIXPDS Reply	46
Sample FIXPDS Reply with D to Display Members	46
Sample DISPLAY Output Showing Member Names.	47
FIXPDS to Delete or Rename a Data Set	47
LISTC Table With UNC to Uncatalog the Data Set	47
Sample LISTC Table with FIX to Invoke FIXPDS	47
Partial FIXPDS Prompt to Update Current Data Set	48
FIXPDS FORMAT1 DSCB Modification Prompt	48
FIXPDS FORMAT1 DSCB DS1DSNAM Update.	48
Sample FIXPDS Update Prompt.	49
Sample FIXPDS Results	49

Chapter 5

Advanced Tips and Techniques	51
Chaining	51
Adding Directory Blocks	53
Adding a Single Extent in Tracks	53
Adding a Single Extent in Cylinders	54
Adding an Entry Point	54
Adding ZAP IDR Records	54
Allocating a Data Set	54
Backing Up Multiple Members	55
Building a Member List	55
Changing Secondary Allocation	57
Checking Alias Members	57
Checkpointing the Log	57

Cloning a Member	58
Combining Multiple Members	58
Comparing Members	58
Comparing Multiple Members	58
Compressing Data Sets	59
Compressing Multiple Data Sets	59
Controlling GO Sessions	59
Controlling Multiple Logs	60
Converting CLIST Members	60
Converting Source Members	60
Copying Members	61
Defining a Dynamic Command	61
Deleting All Members	62
Deleting Multiple Members	62
Disassembling a Member	62
Displaying External References	62
Editing Multiple Members	63
Excluding Multiple Members	63
Executing in Background	63
Executing in CLIST Mode	64
Formatting Data Set VTOC Information	64
Formatting Directory Entries	64
Formatting Member Data	65
Listing Module History	65
Locating Data in Dialog Tables	65
Managing Data Set Lists	66
Managing Member Lists	67
Manipulating Non-Partitioned Data Sets	68
Migrating Test Modules to Production	68
Modifying AMODE and RMODE	69
Modifying DCB Parameters	69
Modifying Member Attributes	70
Modifying Several Members' Attributes	70
Getting Help	71
Processing Multiple Data Sets	71
Reblocking Source Members	72
Reconstructing Linkage-Edit JCL	72
Releasing Space in a Data Set	72
Releasing Space in Multiple Data Sets	73
Renaming Multiple Members	73
Repairing a CSECT in Multiple Members	73
Restoring Deleted Members	73
Saving a LISTC Table	74
Saving a MEMLIST Table	74
Searching a Data Set	74
Searching for Panel, Message, or CLIST Members	75
Searching for System Modules	75
Searching Multiple Data Sets	75

Selecting Members	76
Specifying a Member Group	76
Switching Modes	78
Trimming Dialog Tables	78
Updating a Data Set	79
Updating Multiple Data Sets	79
Validating Data	80
Zapping a CSECT	80
Glossary	81
Index	85

Welcome to StarTool FDM

About This Book	7
Before You Begin	7
Conventions	7
Documentation	9

About This Book

This document describes StarTool[®] FDM (File and Data Manager). It provides reference information concerning advanced system features and supporting operating system calls used by StarTool FDM.

StarTool FDM StarTool FDM is a multi-purpose file and data management utility for IBM[®] mainframe systems. It provides an ISPF-based, menu-driven, integrated interface to a variety of editors and data management tools for PDS, PDSE, VSAM, direct-access, IMS, and Db2 files. A TSO command-line interface is also supported, as well as bulk file processing in batch mode. Customizable option tables and exits make StarTool FDM highly flexible and adaptable to user needs.

Audience This document is intended for use by z/OS mainframe systems programmers and software developers familiar with z/OS system internals.

Before You Begin

New Information Change bars in the left margin (shown at left) identify substantive changes to this publication since StarTool FDM Version 7.6.3.

Corrections and Technical Support The Readme file contains updates and corrections to this manual issued after the publication date. It also provides contact information for Micro Focus Customer Support.

Conventions

Terminology Throughout this document:

- z/OS refers to the z/OS[™] and OS/390[®] IBM[®] operating systems.
- StarTool FDM may also be referenced as StarTool or FDM.

Typographic Conventions

The following textual conventions are used throughout this document to highlight special information:

Convention	Meaning
Bold	Panel title or field name.
<i>Italics</i>	Introduces new terms, sets off important information, or marks document titles.
UPPERCASE	Indicates keys or key combinations; for example, the ENTER key.
Bright blue	Clickable cross-reference or active hyperlink.
Monospaced	JCL, source code, or message text. Also used for member names, file names, and commands if these are not clear from context.
MONOSPACED UPPERCASE	Required value or literal in code or JCL parameter.
monospaced lowercase	<p>Pattern for a field value or parameter you specify. Number of characters is significant. Upper-case characters are literals. Lower-case characters are placeholders that indicate data type, where:</p> <ul style="list-style-type: none"> y = year m = month d = day a = alphanumeric n = numeric x = other or mixed ? = one-character wild card * = n-character wild card <p>Punctuation other than wild cards must be reproduced in the position shown.</p>
<p><i>Examples:</i></p> <ul style="list-style-type: none"> ■ yyyy/mm/dd ■ C'aa' ■ B'nn' ■ D'nn' ■ X'nn' 	<p><i>Examples:</i></p> <ul style="list-style-type: none"> ■ International calendar date with four-character year, two-character month, and two-character day separated by required slashes, such as 2010/01/01. ■ Alphanumeric character string in user-readable form, two characters long, such as C'K9' ■ Binary number, two digits long, where n = 0 to 1, such as B'10' ■ Decimal number, two digits long, where n = 0 to 9, such as D'10' ■ Hexadecimal number, two digits long, where n = 0 to F, such as X'C1'
<i>monospaced italics</i>	Descriptive placeholder for value or parameter you specify, but not a pattern; for example, <i>filename</i> .
Square braces []	Optional parameter or choice of values. May be nested.
Vertical bar	Inside braces, a vertical bar separates mutually exclusive parameter choices or values.
Ellipsis ...	Optional repetitions of a pattern in a list.

Convention	Meaning
Greater-than symbol >	Separates items in a chain of menu or command selections on a GUI client. For example, Start > All Programs > Micro Focus > <i>product_name</i> .

Documentation

All StarTool FDM documentation is provided in Adobe® Portable Document Format (PDF) and may be downloaded from the following Micro Focus website:

<https://www.microfocus.com/support-and-services/documentation/>

No user ID or password is needed to access the documentation.

Related Publications

Available StarTool FDM publications include:

Title	Description
<i>StarTool FDM Installation Guide</i>	System requirements, installation instructions, and configuration information for StarTool FDM.
<i>SER10TY User's Guide</i>	Installation information for SER10TY licensing software and instructions on how to apply license key SERTificates.
<i>StarTool FDM Quick Reference</i>	Overview of StarTool FDM commands, with syntax details for frequently used functions. Includes PEDIT and StarBat subcommands.
<i>StarTool FDM User's Guide</i>	StarTool FDM concepts and facilities, with instructions for using the ISPF-based menu-driven interface.
<i>StarTool FDM Command Reference</i>	TSO command-line syntax and parameter reference, organized alphabetically. Interactive subcommands included for major functions.
<i>StarTool FDM System Services</i>	Advanced reference to operating system calls used by StarTool FDM.
<i>StarTool FDM StarBat Option</i>	Batch-mode interface for bulk changes to data sets. StarTool FDM functions invoked by JCL procedures.
<i>StarTool FDM Db2 Option</i>	StarTool FDM data management functions for Db2 relational database tables, columns, and rows, with SQL processing support.
<i>StarTool FDM IMS Option</i>	StarTool FDM data management functions for IMS hierarchical database files and structures.
<i>StarTool FDM Extended Compare Option</i>	Integrated file comparison utility based on Micro Focus Comparex. Data file versus text file comparison logic.
<i>StarTool FDM Messages</i>	Consolidated message reference for base product and all licensed product options, with recovery recommendations.

Using the PDF Documentation

To view PDF files, use Adobe® Reader®, which is freely available from Adobe on the World Wide Web at <http://www.adobe.com>. Reader Version 7.0.5 or higher is recommended.



TIP Be sure to download the *full version* of Reader. The more basic version does not include the cross-document search feature.

This section highlights some of the main Reader features. For more detailed information, see the Adobe Reader online help system.

The PDF manuals include the following features:

- **Bookmarks.** All of the online manuals contain predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left of each online manual.
- **Links.** Cross-reference links within an online manual enable you to jump to other sections within the manual and to other manuals with a single mouse click. These links appear in blue.
- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.
- **Advanced search.** Starting with Version 6, Adobe Reader includes an advanced search feature that enables you to search across multiple PDF files in a specified directory. (This is in addition to using any search index created by Adobe Catalog—see step 3 below.)

To search across multiple PDF documents at once, perform the following steps (requires Adobe Reader Version 6 or higher):

- 1 In Adobe Reader, select Edit > Search (or press CTRL+F).
- 2 In the text box, enter the word or phrase for which you want to search.
- 3 Select the **All PDF Documents in** option, and browse to select the folder in which you want to search. (If you have a document open that has an Adobe Catalog index attached, you can leave the **In the index named...** option selected to search across all the manuals in the index.)
- 4 Optionally, select one or more of the additional search options, such as **Whole words only** and **Case-Sensitive**.
- 5 Click the **Search** button.



NOTE Optionally, you can click the **Use Advanced Search Options** link near the lower right corner of the application window to enable additional, more powerful search options. (If this link says **Use Basic Search Options** instead, the advanced options are already enabled.) For details, see Adobe Reader's online help.

Chapter 1

Advanced Options Menu

The **Advanced Options** menu provides access to certain options that are specifically used by systems programmers. Only these advanced options are covered in this section. For information about other options of the **Advanced Options** menu, refer to the *StarTool FDM User's Guide*.

Advanced Options Menu

Access the **Advanced Options** menu by selecting option A from the **Primary Options** menu. The following menu displays.

```
----- 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 G0 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 CBSelect     - Use copybooks to specify record selection when copying
  18 CBSelect     - 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 fields 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 selected automatically 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 fields 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 members. 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.
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 Micro Focus Comparex [®] product, which is separately licensed.
17 - CBSelect	Copies selected records using copybook variable names.
18 - CBSelect	Use a saved copybook selection table to copy selected records

Examine Environment Settings

From the **Advanced Options** menu, select **Option 13 - Examine**, to:

- locate system 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

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

```

----- StarTool Examine Current Environment -----
OPTION  ==>

Choose one of the following:

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

```

The options on this menu are:

FINDMOD	searches for system modules. You can examine redundant copies of members in the LINKLIST or LPALIST.
LISTA	displays allocated DDNAMEs. You can filter the displayed DDNAMEs on several different criteria.
DDNAME	displays allocated DDNAMEs filtering by a mask name.
LISTV	displays disk volume space and attributes. You can filter the displayed volumes using a volume name mask.
SVCMAP	displays system SVCs from the executing operating system.
WHOHAS	displays users allocated to a data set.
STATUS	displays the current status of all StarTool FDM ISPMODE functions. You can switch to any other function from STATUS.
LISTENV	displays the current software and hardware environment. Use the CONTROL subcommand with a LISTENV operand.
DEFAULTS	displays StarTool FDM customized defaults. Use the CONTROL subcommand with a DEFAULTS operand to display options selected in the PDS#OPT4 module.
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.
CATALOGS	displays current active system catalogs. Use the CAX function.

SYSTAT	invokes a software product called Sys/Stat from Advent Software if available at your installation.
VUSE	displays disk volume usage, space and statistics.

Current Hardware and Software Environment

From the Examine Current Environment menu, select option **8 or LISTENV** to display the current software and hardware environment. For example:

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT ----->Control listenv Columns 00001 00072
***** ***** Top of Data *****
000001
000002 PDS100I START00L/DB-IM-SB-XC-SW -- Version v.r.m
000003
000004 PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER,
      TRANSLATO
000005 PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
000006 PDS031I Input buffering: RETAIN(9)
000007 PDS036I Largest free storage area is 6052K
000008 PDS046I Largest area above the line is 1805M
000009
000010 PDS280I System serial:540BA; CPU type:2066
000011 PDS281I Active CPUs:0 1
000012 PDS282I SMF ID:Q001; System mode:ESAME
000013 PDS283I Maintenance data:SP7.0.3 HBB7706
000014 PDS284I IPL date:2007/04/10 2007.100; Time:02:13
000015 PDS285I IPL type:CLPA; Volume:Z3RES1; UNIT:4BC
000016 PDS286I Master catalog dsname:CATALOG.OS390.MASTER; Volume:OS39M1;
      UNIT:
000017 PDS287I z/OS 1.3.0;; DFSMS 1.3.0; ISPF 5.2;
000018 PDS287I VTAM 6.1.2; TSO/E 3.03.0; RACF 7.0.6;
000019 PDS288I Current NUCLEUS ID:1; I/O CONFIG ID:A0
Command ==> Scroll ==> PAGE

```

StarTool FDM identifies the hardware and software environment with a display of the different types of system storage. Areas that are not present in a system are not shown on the panel. In this example, an FLPA area was not established, but an E-FLPA area was established and is described.

This is an excerpt from a log created with LISTENV:

```

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:2007/04/14 2007.084; 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; DFSMSHsm; 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    042CAFFF      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    00FFFFFF      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    007FFFFFFF     8,172K
PDS059I V=R AREA             00005000    00024FFF      128K
PDS059I SYSTEM               00001000    00004FFF       16K
PDS059I PSA                   00000000    00000FFF        4K

```

Customized Defaults

From the **Examine Current Environment** menu, select option **9 or DEFAULTS** to display the current StarTool FDM default configuration settings. StarTool FDM displays the installation parameters chosen when you installed StarTool FDM. These include:

- Current product version, release, modification level, and build
- Global operands
- Input buffering
- Largest free storage areas
- Currently active version of the user-customizable options module, PDS#OPT4.

This information is often useful in debugging environmental problems in StarTool FDM.

For example:

```
FUNCTIONS CONTROL DSN CMDS MEM CMDS A-M MEM CMDS N-Z DEFAULTS FEATURES
>----->Control defaults

PDS100I PDSE/SuperEdit -- Version v.r.m

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 2007/04/14 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
```

Chapter 2

CLIST Processing

There are multiple ways to start a CLIST in StarTool FDM.

External and Internal CLISTs

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.

Using CLISTs in ISPMODE

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.

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.

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

- DCONTROL controls how to handle displays and their responses. 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
```

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

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 skips 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. 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 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. The 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.

DSETVAR Command

Syntax

DSETVAR *internal-variable*|VAR(*ISPF variable*) '*character string*'

<i>internal variable</i>	Is one of the following values:	
	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(<i>name</i>)	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.	
<i>'character string'</i>	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

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.

DGETVAR Command

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.

Syntax

DGETVAR *internal-variable*|VAR(*ISPF variable*) *CLIST-variable*

<i>internal-variable</i>	Is one of the following:	
	ZCMD	command line input variable
	PDSZSEL	command variable from prompt panel
	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: <ul style="list-style-type: none"> ■ 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(<i>name</i>)	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.	
<i>CLIST-variable</i>	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

<i>panel-name</i>	Name of an ISPF panel to display. This can be a StarTool FDM panel or any other panel accessible from the ISPF environment.
<i>blank</i>	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 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.

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
```

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#B1AD) '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 pull-down 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(DUMMYMEM)'"
DSET VAR(PDSPDISP) 'S'      ; /* DATA SET DISPOSITION */
DSET VAR(PDSPHRI) '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

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
IF &STAT=I THEN EXIT      ;/* can not build the table
DGET FUNC STAT
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

```

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.

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
```


Chapter 3

Library Lookaside Processing

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

LLA-Managed Libraries

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.

Chapter 4

FIXPDS Subcommand

The FIXPDS subcommand may be used to modify PDS attributes of the the current (or *active*) data set. But 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. These, too, can be modified using the FIXPDS subcommand.

Modifying PDS Attributes for an Active Data Set

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.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----
>----->c lib.cntlr
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
DS200I SHR  3380 C   FB      80   1600   1X   10      9   10 TRK    8

PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/
***** Bottom of data *****
```

Sample FIXPDS Promp

```
----- Reply Required ----- Enter a prompt response
COMMAND ==> yes                               SCROLL ==> CSR
Reply required -- Enter YES or NO
- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----
>----->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) ?
***** Bottom of data *****
```

Sample FIXPDS Result

```

----- ISPMODE Session# 1 Log# 1 ----- Row 82 to 92 of 92
COMMAND ==>
                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----
>----->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 *****

```

Sample FIXPDS to Add 10 Disk Tracks

```

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

- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----

Change the directory:
Add directory blocks ==> (1 to 9999)
Percent free blocks ==> (1 to 999)
Release disk space:
Release all free space ==> NO
Release any free extents ==> NO
Percent of data set free ==> (1 to 999)
Add a disk extent:
Add disk tracks ==> 10
Add disk cylinders ==>
Percent of data set free ==> (1 to 999)
Modify data set attributes:
Management class ==>
Storage class ==>
Expiration date ==>
Secondary space type ==> TRK
Secondary amount ==> 10
Modify DCB parameters:
Set DCB LRECL ==> 80
Set DCB BLKSIZE ==> 1600
Set DCB RECFM ==> FB
Set DCB OPTCD ==> C
Modify other parameters:
Reset directory entries ==>
Check for EDIT sessions ==> NO

Total directories: 10
used: 2
Total tracks: 10
free: 9
(migration and backup attributes)
(storage unit, volume, storage service)
(Julian -- yyyyddd)
(BLK, CYL, TRK or RNDBLK)
(0 to 99999 in space type units)
(1 to 32767)
(1 to 32767)
(F/FB/FA/FM/FBA/FBM/V/VB/VBA/VBM/U)
(W, C, Z, WC, WZ, CZ, WCZ or NO)
(1 to 9999; use * for the same number)
(YES or NO for use with Reset)

```

Changing the FORMAT1 DSCB for an Alternate Data Set

FIXPDS supports changing fields in the FORMAT1 DSCB directly if you set your Interface level to Advanced or Power in the **SETALL** option. This allows you to modify the PDS attributes of an *alternate* data set — that is, a data set that is not currently open.

The sample panels illustrate this feature. Actual panels will differ somewhat from those shown.

SETALL Panel to Change Interface Level

```

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

- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----

Change the directory:                                Total directories:    10
  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 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 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)

```

FIXPDS Subcommand

```

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

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 *****

```

FIXPDS to Update Current Data Set

```

OPTION ==>

- DSN=WSER07.LIB.CNTRL,VOL=SER=SER001 MEM=PDSDCB/ -----
                                                    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)

```

FORMAT1 DSCB Prompt

```

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

Current - DSN=WSER07.LIB.CNTRLR,VOL=SER=SER001 MEM=PDS DSCB/ -----
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.CNTRLR
(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 ==> 02 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 ==> 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 ==> 010100AC000A00AD0004 Extent 2 in XX01CCCCHHHHCCCCHHHH
(7D,10) DS1EXT3 ==> 00000000000000000000 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=PDS DSCB/ -----
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.CNTRL,VOL=SER=SER001 MEM=PDS DSCB/ -----
DSCB1 at 000A000017

                                                    More:  -
(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 XX00CCCCHHHHCCCCHHHH
(73,10) DS1EXT2 ==> 010100AC000A00AD0004 Extent 2 in XX01CCCCHHHHCCCCHHHH
(7D,10) DS1EXT3 ==> 00000000000000000000 Extent 3 in XX02CCCCHHHHCCCCHHHH
(87,6) DS1PTRDS ==> 0000000000    CCHHR of any associated Format 3 DSCB

```

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.CNTRL,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) ?
***** Bottom of data *****

```

Sample FIXPDS DSCB Update Results

```

----- ISPMODE Session# 1 Log# 1 ---- Row 93 to 103 of 103
COMMAND ==>
                                SCROLL ==> CSR
- DSN=WSER07.LIB.CNTRL,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

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

```

Sample FIXPDS to Modify an Alternate Data Set

```

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

```

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:                                       free:      718
  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         ==>      (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.cntl (required if not cataloged)
  Alternate volume       ==>

```

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

```

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=PDSDCB/ -----
>----->Fixpds DSCB(5E 8000000A C0000001) MODDSNAME('WSER07.LIB.CNTLS')
MODVOL
-----UME(SER001)
PDS398A Should this alternate data set be modified (Yes/No)
?***** Bottom of data
*****
```

Sample FIXPDS Reply

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

Sample FIXPDS Results

```

----- ISPMODE Session# 1 Log# 1 ----- Row 72 to 76 of 76
COMMAND ==>                               SCROLL ==> CSR
- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001 MEM=PDSDCB/ -----
>----->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=PDSDCB/
***** Bottom of data *****

```

Restoring a Deleted Data Set

StarTool FDM restores 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.CNTLO_____
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
-----
EDIT
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=IEFBR14
000004 //SYSLIB DD DSN=WSER07.LIB.RESTORE,DISP=(NEW,CATLG),VOL=SER=WORK01,
000005 // UNIT=SYSALLDA,SPACE=(ABSTR,(200,1180))
***** ***** Bottom of Data *****

```

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.VTOCIX.WORK01
000A0000 000B000E 1 *---30 *** FORMAT 4 (VTOC) 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.

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 *****

```

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 *****

```

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.

LISTC Table With UNC to Uncatalog the Data Set

```

----- 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_____ 0539H2 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
***** Bottom of data *****

```

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_____ 0539H2 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 *****

```

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

```

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

```

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

```

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 NEWSNAME('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 NEWSNAME('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 *****

```


Chapter 5

Advanced Tips and Techniques

StarTool FDM has many subcommands. However, there are other ways to perform tasks associated with these subcommands. The advanced tips and techniques in this chapter suggest alternatives and options, with examples to show how subcommands are related to each other.

Chaining

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, tasks are identified 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	Minimum Level
"Adding Directory Blocks"	Standard
"Adding a Single Extent in Tracks"	Standard
"Adding a Single Extent in Cylinders"	Standard
"Adding an Entry Point"	Advanced
"Adding ZAP IDR Records"	Advanced
"Allocating a Data Set"	Standard
"Backing Up Multiple Members"	Advanced
"Building a Member List"	Standard
"Changing Secondary Allocation"	Standard
"Checking Alias Members"	Advanced
"Checkpointing the Log"	Standard
"Cloning a Member"	Standard
"Combining Multiple Members"	Advanced
"Comparing Members"	Standard
"Comparing Multiple Members"	Power

Task	Minimum Level
"Compressing Data Sets"	Standard
"Compressing Multiple Data Sets"	Advanced
"Controlling GO Sessions"	Power
"Controlling Multiple Logs"	Power
"Converting CLIST Members"	Advanced
"Converting Source Members"	Advanced
"Copying Members"	Standard
"Defining a Dynamic Command"	Advanced
"Deleting All Members"	Power
"Deleting Multiple Members"	Standard
"Disassembling a Member"	Advanced
"Displaying External References"	Advanced
"Editing Multiple Members"	Standard
"Excluding Multiple Members"	Standard
"Executing in Background"	Power
"Executing in CLIST Mode"	Power
"Formatting Data Set VTOC Information"	Advanced
"Formatting Directory Entries"	Advanced
"Formatting Member Data"	Standard
"Listing Module History"	Standard
"Locating Data in Dialog Tables"	Standard
"Managing Data Set Lists"	Standard
"Managing Member Lists"	Standard
"Manipulating Non-Partitioned Data Sets"	Advanced
"Migrating Test Modules to Production"	Standard
"Modifying AMODE and RMODE"	Standard
"Modifying DCB Parameters"	Advanced
"Modifying Member Attributes"	Advanced
"Modifying Member Statistics"	Standard
"Modifying Several Members' Attributes"	Advanced
"Getting Help"	Standard
"Processing Multiple Data Sets"	Standard
"Reblocking Source Members"	Advanced
"Reconstructing Linkage-Edit JCL"	Advanced
"Releasing Space in a Data Set"	Standard

Task	Minimum Level
"Releasing Space in Multiple Data Sets"	Advanced
"Renaming Multiple Members"	Power
"Repairing a CSECT in Multiple Members"	Advanced
"Restoring Deleted Members"	Standard
"Saving a LISTC Table"	Standard
"Saving a MEMLIST Table"	Standard
"Searching a Data Set"	Standard
"Searching for Panel, Message, or CLIST Members"	Advanced
"Searching for System Modules"	Advanced
"Searching Multiple Data Sets"	Standard
"Selecting Members"	Standard
"Specifying a Member Group"	Standard
"Switching Modes"	Advanced
"Trimming Dialog Tables"	Standard
"Updating a Data Set"	Standard
"Updating Multiple Data Sets"	Standard
"Validating Data"	Standard
"Zapping a CSECT"	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 an external symbol, the new alias is assigned the entry point of that symbol.

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**



NOTE The MODEL command does not allow allocation of multivolume data sets.

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

Combination	<p>Example: MEMLIST start*end AB*C selects all members whose names start with AB and contain C at the end of the member name.</p> <p>ABC* selects all members whose names start with ABC.</p> <p>*ABC selects all members containing ABC at the end of the member name.</p> <p>You can use placeholders (% or ?) that match any single character anywhere in a combination specification.</p>
Placeholder	<p>Example: MEMLIST start%end A%?C selects all members with four-character names beginning with A and ending with E.</p> <p>A% selects all members with two-character names beginning with A.</p> <p>?F selects all members with two-character names ending with F.</p> <p>You can use placeholders (% or ?) that match any single character anywhere in a pattern or combination specification.</p>

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.

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 any 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 included automatically in a copy. In addition, if COPYMOD reblocking of load members is required due to data set block size differences, a COPYMOD operation is requested for the block size 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 only be copied to the output data set if a member with that name is already there. Use this parameter 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. 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 also deletes associated (alias) members. If you want to delete load members beginning with *ABC*, type either of the following to clean up any associated members:

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

If you type a subcommand to delete multiple members, the **DELETE** subcommand displays the members that are affected and prompts for continuation.

Disassembling a Member

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

For the menu system, type:

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 processes 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 is 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 and performed in a batch process (see ["Executing in Background" on page 63](#)). 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'  
. . .
```

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. Type DIRENTRY as a primary command for any desired member group. Type 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 (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. Type 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, type *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 with the SAVE and MERGE commands. MEMLIST table names have from one to six alphanumeric characters.

Specify member ranges by using a starting member name followed by a colon (:) and an ending member name. To limit the member list displayed, use a selected range as follows: ML ABC:BCD RESET. This 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.

You can also build a member list whose member names contain a common string. For example, ML IST/ builds a member list containing all members with the character string IST somewhere in the member name.

Other 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 way to obtain member lists. The criteria varies with over 80 selectable options from linkage-edit date and module attributes to SSI and IDR data in the members. This expands the ease in identifying the members you want to manipulate. See *Appendix A, Formatting Member Names*, in the StarTool® FDM Reference Guide for member group specification rules and *Chapter 6: Commands – L to N* in the Reference Guide, for the MEMLIST function about 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 type 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 that 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
- 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:

<ul style="list-style-type: none"> ■ In the batch JCL, allocate TEST, PROD and BACKUP files as follows: <pre>//TEST DD DSN=test.library,DISP=OLD //PROD DD DSN=production.library,DISP=SHR //BACKUP DD DSN=backup.library,DISP=SHR</pre> 	
■ 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/AMODE64/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 the attributes of the load member is with the ATTR line command from a MEMLIST table. This displays a prefilled entry assist panel that reflects the current attributes of the member. Change individual attribute entries as desired.

Modifying Member Statistics

Modify a source member's ISPF statistics 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.

For 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
```

Getting Help

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 (PF6 or PF18).
- For a directory of available commands, type O (OPTIONS) as a line command or primary command. This displays a syntax assist panel for the command. 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 (MENU) as a line command or primary command. This displays 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 a syntax assist or entry assist panel, press HELP for a subcommand tutorial.

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 59 for more information.
GLOBAL	See "Releasing Space in a Data Set" on page 72 for more information.
FIND	See "Searching Multiple Data Sets" on page 75 for more information.
MODEL	Changes 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 79 on page Updating Multiple Data Sets for more information.
SEEK	Changes 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 block size 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 block size 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 the 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. 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. An 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 point in the table (by scrolling or the F command) then type 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 with 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.

To recall a previously saved table, type 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 to 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 76](#)).

For the **IF** subcommand, type attributes next (for example, type 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, type 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.

Other parameters can be in any order following the string specification. 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	<p>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.</p>
Pattern	<p>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.</p>
Combination	<p>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.</p>
Placeholder	<p>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.</p>
Asterisk	<p>Example: MEMBER * Use a single asterisk to refer to the current member group.</p>
Equal	<p>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.</p>

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, type mode switch commands such as 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). For example, 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 type 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 with the END command. Type 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 type 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.

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 for 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 type an X primary command to trim the entire table. Finally, type 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. Type 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, type:

```
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, type a command similar to:

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

From the command line in that table, type 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 type 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, type VERIFY A:B (where A:B is the group of members to be checked).

To validate an entire PDS, type VERIFY `:`

For the menu system, type:

```
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, type 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.

Glossary

A

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.

B

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 <u>are</u> 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.

C

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.

D

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.

E

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

F

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

I

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.
----------------	---

L

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.

M

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.

O

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

P

PDS or Partitioned data set	PDS. A data set with fixed, variable or undefined format that is used as a <i>library</i> 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.

Q

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

S

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 IBM mainframe 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.

T

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.

Index

A

- active data set 33
- ADDCYL 54
- ADDDIR 53
- adding
 - a single extent in cylinders 51
 - a single extent in tracks 51
 - an entry point 51
 - cylinders, single extent 54
 - directory blocks 51, 53
 - entry point 54
 - tracks, single extent 53
 - ZAP
 - IDR records 51, 54
- ADDTRK 54
- advanced options
 - described 12
- Advanced Options menu 11
- ALIAS
 - operand 58
 - subcommand 54
- ALIASCHK command 57
- allocating
 - a data set 51, 54
- alternate data set 35
- AMODE
 - modifying 69
- asterisk entry type 77
- ATTR line command 69
- ATTRIB
 - subcommand 54, 70

B

- background
 - executing in 63
- backing up
 - multiple members 51, 55
- batch mode 63
- building
 - a member list 51
 - See creating

C

- chaining
 - multiple menu levels 51
- CHANGE operand 58

- changing
 - secondary allocations 51
- CHECK keyword 62
- checking
 - alias members 51
- checkpointing
 - the log 51
- CLIST categories
 - external 17
 - internal 17
- CLIST members
 - converting 60
 - searching 75
- CLIST mode 64
- CLIST processing 17
- CLISTS
 - in batch 17
 - in ISPMODE 17
 - interactively 17
 - invoking with CMD command 61
 - processing guidelines 28
- cloning a member 51
- CMD 61
- combination entry type 77
- COMBINE subcommand 58
- combining multiple members 51
- commands
 - ALIASCHK 57
 - DCONTROL 18
 - DGETVAR 18
 - DSETVAR 18
 - EDITLOG 73
 - EXCLUDE 63, 78
 - GLOBAL 71
 - L 65
 - LOCATE 65
 - MODEL 54
 - PANEL 18
 - SEEK 75
 - SETUSER 61
 - UT 61
 - X 63, 78
- COMP line command 58
- COMPARE subcommand 58
- comparing
 - members 51
 - multiple members 51
- COMPDIR subcommand 58
- COMPRESS
 - command 59
 - global command 71

- subcommand 59
- compressing
 - data sets 52
 - multiple data sets 52
- CONDEND subcommand 63
- controlling
 - GO sessions 52
 - multiple logs 52
- converting
 - CLIST members 52, 60
 - source members 52, 60
- COPY
 - command 61
 - subcommand 61
- copying
 - members 52
- creating
 - member list 55
- CSECTs
 - function 80
 - repairing in multiple members 73
 - switching 78
 - zapping 80
- cylinders
 - adding single extent 54

D

- data
 - validating 80
- data set
 - allocating 54
- data set lists 66
- data set repair 35, 43, 47
- data set space
 - releasing 72
- data sets
 - compressing 59
 - compressing multiple 59
 - direct 68
 - mutiple, processing 71
 - searching 74
 - sequential 68
 - updating 79
 - updating multiple 79
 - VSAM 68
- DCB parameters
 - modifying 69
- DCONTROL
 - command 18
- DDNAME
 - function 75
- defining
 - a dynamic command 52, 61
- DELETE

- deleting multiple members 62
 - subcommand 62
- deleted members
 - restoring 73
- deleting
 - all members 52
 - multiple members 52
- DGETVAR
 - command 18, 21
- dialog tables
 - locating 65
 - trimming 78
- direct data sets 68
- directory blocks
 - adding 53
- directory entries
 - formatting 64
- DIRENTRY
 - subcommand 64
- DISASM
 - subcommand 62
- disassembling members 52, 62
- displaying
 - external references 52, 62
- DSETVAR
 - command 18, 21
- DUP subcommand 60

E

- Edit macro
 - ADDCYL 54
 - ADDDIR 53
 - ADDTRK 54
- EDIT subcommand 63
- editing
 - multiple members 52, 63
- EDITLOG
 - command 28, 73
- entry point
 - adding 54
- entry types
 - asterisk 77
 - combination 77
 - equal 77
 - pattern 77
 - placeholder 77
 - range 77
- equal entry type 77
- EXCLUDE
 - command 63, 78
 - copying members 61
 - subcommand 56, 63, 78
- excluding multiple members 52, 63
- executing 64

- batch mode 63
- in background 52
- in CLIST mode 52

- EXIST 61
- express mode 20
- external references
 - displaying 62

F

- FIND
 - global command 71, 75
 - subcommand 56, 65, 74, 76
- FINDMOD subcommand 75
- FIXPDS 35, 47, 72
 - subcommand 53, 57, 62, 69, 72, 73
- formatting
 - data set VTOC information 52
 - directory entries 52, 64
 - member data 52, 65
- formatting VTOC information 64
- functions
 - CSECTS 80
 - DDNAME 75
 - LC 66, 74
 - LF 66, 74
 - LISTC 66, 74
 - LISTF 66, 74
 - MEMLIST 55, 67, 74
 - ML 55, 67, 74
 - ZAP 80

G

- getting help 71
- GLOBAL 71
 - command 71, 73
 - global command 71, 73
- global commands
 - COMPRESS 71
 - FIND 71, 75
 - GLOBAL 71, 73
 - MODEL 71
 - REPLACE 71, 79
 - SEEK 71
 - WHOHAS 71
- GO
 - controlling sessions 59

H

- help 52, 71
- HISTORY subcommand 65

I

- IF subcommand 56, 73, 76
- information
 - module history 65
 - See *also* translator data 65
 - See *also* USERDATA information 65
 - See *also* Zap information
- ISPF panel applications
 - invoking with XPANEL 61
- ISPF program applications
 - invoking with PGM command 61
- ISPMODE
 - CLISTs 17
 - using CLISTs in 17

L

- L Command 65
- LC Function 66, 74
- LF Function 66, 74
- libraries
 - LLA managed 31
- linkage-edit JCL, reconstructing 72
- LIST
 - subcommand 65
- list member specification 76
- LISTC
 - function 66, 74
 - primary command 66
 - saving table 74
- LISTF Function 66, 74
- listing module history 52
- LLA managed libraries 31
- LOCATE command 65
- locating
 - data in dialog tables 52
 - dialog tables 65
- log
 - checkpointing 57
- logs
 - controlling multiple 60

M

- managing
 - data set lists 52, 66
 - member lists 52, 67
- manipulating
 - data sets 68
 - non-partitioned data sets 52
- MAP subcommand 69, 72, 73
- member attributes
 - modifying 70

- member data
 - formatting 65
- member group
 - specifying 76
- member lists
 - creating 55
 - managing 67
- member statistics
 - modifying 70
- members
 - cloning 58
 - combining multiple 58
 - comparing 58
 - comparing multiple 58
 - copying 61
 - deleted 31
 - deleting all 62
 - deleting multiple 62
 - disassembling 62
 - restoring deleted 73
 - selecting 76
- MEMLIST
 - function 55, 67, 74
 - saving table 74
 - switching 78
- messages
 - PDS#102M 20
 - searching 75
- migrating
 - test files 68
 - test modules to production 52
- ML function 55, 67, 74
- mode
 - switching 78
- MODEL
 - command 54
 - global command 71
- modes
 - batch 63
 - CLIST 64
 - express 20
 - switching 78
- modifying
 - AMODE 69
 - AMODE and RMODE 52
 - DCB parameters 52, 69
 - member attributes 52, 70
 - member statistics 52, 70
 - RMODE 69
 - several members' attributes 52
- module history 65
- multiple data sets
 - updating 79
- multiple members
 - backing up 55
 - editing 63

- excluding 63
- renaming 73
- repairing CSECTs 73

N

- NOCHANGE operand 58
- NOEXIST 61
- non-partitioned data sets 68

O

- obtaining
 - translator data 65
 - USERDATA information 65
 - ZAP information 65
- operands
 - DCONTROL 19
 - DGETVAR 22
 - DSETVAR 20
 - PANEL 23
- options
 - advanced 12
 - current hardware and software environment 14
 - customized defaults 15

P

- PANEL
 - command 18
 - command syntax 22
- panels
 - Advanced Options 11
 - searching 75
- pattern entry type 77
- pattern member specification 76
- PDS#102M 20
- PGM 61
- placeholder entry type 77
- placeholder member specification 76
- processing multiple data sets 52, 71
- production
 - migrating files to 68

R

- range
 - member specification 76
- range entry type 77
- reblocking source members 52, 72
- reconstructing linkage-edit JCL 52, 72

- releasing
 - data set space 72
 - space in a data set 52
 - space in multiple data sets 52
 - RENAME subcommand 73
 - renaming multiple members 53, 73
 - repairing
 - CSECTs in multiple members 53, 73
 - REPLACE 61
 - global command 71, 79
 - subcommand 65, 79
 - replacing existing files
 - EXIST command 61
 - replacing existing members
 - REPLACE command 61
 - REPRO
 - subcommand 54, 72
 - REPRO line command 58
 - REPRO Subcommand 55
 - REPRO subcommand 55
 - RESTORE subcommand 73
 - restoring
 - deleted data sets 43
 - deleted members 53, 73
 - REXX 17
 - RMODE
 - modifying 69
- S**
- saving
 - LISTC table 53, 74
 - MEMLIST table 53, 74
 - searching
 - a data set 53
 - CLIST members 75
 - data sets 74
 - for panel, message or CLIST members 53
 - for system modules 53
 - messages 75
 - multiple data sets 53, 75
 - panels 75
 - system modules 75
 - SEEK
 - command 75
 - global command 71
 - selecting members 53, 76
 - SEPARATE subcommand 58
 - sequential data sets 68
 - SETUSER command 61
 - short-cut commands 56
 - source members
 - converting 60
 - reblocking 72
 - specifying
 - member group 53, 76
 - starting StarTool FDM
 - subcommands 61
 - subcommands 51
 - ALIAS 54
 - ATTRIB 54, 70
 - COPY 61
 - DELETE 62
 - DIRENTRY 64
 - DISASM 62
 - DUP 60
 - EDIT 63
 - EXCLUDE 78
 - FIND 65, 74, 76
 - FINDMOD 75
 - FIXPDS 53, 57, 62, 69, 72, 73
 - HISTORY 65
 - IF 73, 76
 - LIST 65
 - MAP 69, 72, 73
 - RENAME 73
 - REPLACE 65, 79
 - REPRO 54, 55, 72
 - RESTORE 73
 - starting 61
 - USAGE 64
 - VERIFY 80
 - XREF 62
 - switching modes 53, 78
 - syntax
 - DGETVAR command 22
 - DSETVAR command 20
 - PANEL command 22
 - system modules
 - searching 75
- T**
- test files
 - migrating to production 68
 - tracks
 - adding single extent 53
 - translator data
 - obtaining 65
 - TRAP 61
 - trimming
 - dialog tables 53
 - member lists 56
 - tables 78
 - TSO commands
 - invoking with CMD command 61
 - invoking with TRAP command 61

U

- updating
 - a data set 53
 - multiple data sets 53, 79
- USAGE subcommand 64
- USERDATA information
 - obtaining 65
- UT command 61

V

- validating data 53, 80
- VERIFY subcommand 31, 80
- VSAM data sets 68
- VTOC
 - formatting information 64

W

- WHOHAS global command 71

X

- X
 - command 63, 78
 - line command 56
 - primary command 56
- XPANEL 61
- XREF subcommand 62

Z

- ZAP
 - adding IDR records 54
 - function 80
 - obtaining information 65
 - switching functions 78
- zapping CSECTs 53, 80