StarTool® FDM

Command Reference

Table of Contents

Welcome To StarTool FDM

About This Book

This document describes StarTool® FDM (File and Data Manager) Version 7.8 TSO command-line syntax and parameters supported by StarTool FDM primary commands, subcommands, and function-specific commands. For reference purposes, this document is organized alphabetically by command name.

The separately licensed StarBat Option, DB2 Option, IMS Option, and Extended Compare Option are described in other manuals in the StarTool FDM documentation suite.

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 file editors and data management tools for PDS, PDSE, VSAM, direct-access, IMS, and DB2 files. A TSO command-line interface is supported concurrently, and bulk file updates in batch mode are also supported. Separately licensed product options allow users to purchase only the capabilities they need.

Audience   The intended audience for this document is IBM mainframe systems programmers and application specialists who are already familiar with StarTool FDM commands and want a reference organized by name for look-up of command syntax and function details.

Before You Begin

New Information   Change bars in the left margin (shown at left) identify substantive changes to this user manual since it was last published in January 2011.

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 IBM® z/OS™ operating system.

- StarTool FDM may also be referenced as StarTool or FDM.

- *somnode* is the high-level qualifier you specify for StarTool FDM installation libraries.

- *vrm* is the version, release, and modification level of StarTool FDM without punctuation; for example, for StarTool FDM Version 7.8, the value of *vrm* is 780.

Typographic Conventions

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

Convention	Meaning
Bold	Panel title or field name.
Italics	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	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: y = year m = month d = day a = alphanumeric n = numeric x = other or mixed ? = one-character wild card * = *n*-character wild card Punctuation other than wild cards must be reproduced in the position shown.
Examples: - yyyy/mm/dd - C'aa' - B'nn' - D'nn' - X'nn'	*Examples:* - 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'
monospaced italics	Descriptive placeholder for value or parameter you specify, but not a pattern; for example, *filename*.
Square braces [ ]	Optional parameter or choice of values. May be nested.

Convention	Meaning
Vertical bar |	Inside braces, a vertical bar separates mutually exclusive parameter choices or values.
Ellipsis . . .	Optional repetitions of a pattern in a list.
Greater-than symbol >	Separates items in a chain of menu or command selections on a Graphic User Interface (GUI) client. For example, **Start > All Programs > Micro Focus > *product_name*.**

Documentation

StarTool FDM product documentation is provided in Adobe Portable Document Format (PDF). You can download a complete set of documentation for StarTool FDM from the Micro Focus Customer Support Web site at https://supportline.microfocus.com/.

StarTool FDM Documentation Suite

Available StarTool FDM publications include:

Title	Description
StarTool FDM Installation Guide	System requirements, installation instructions, and configuration information for StarTool FDM.
StarTool FDM Quick Reference	Overview of StarTool FDM commands, with syntax details for frequently used functions. Includes PEDIT and StarBat subcommands.
StarTool FDM User's Guide	StarTool FDM concepts and facilities, with instructions for using the ISPF-based menu-driven interface.
StarTool FDM Command Reference	TSO command-line syntax and parameter reference, organized alphabetically. Interactive subcommands included for major functions.
StarTool FDM System Services	Advanced reference to operating system calls used by StarTool FDM.
StarTool FDM StarBat Option	Batch-mode interface for bulk changes to data sets. StarTool FDM functions invoked by JCL procedures.
StarTool FDM DB2 Option	StarTool FDM data management functions for DB2 relational database tables, columns, and rows, with SQL processing support.
StarTool FDM IMS Option	StarTool FDM data management functions for IMS hierarchical database files and structures.
StarTool FDM Extended Compare Option	Integrated file comparison utility based on Micro Focus Comparex. Data file versus text file comparison logic.
StarTool FDM Messages	Consolidated message reference for base product and all licensed product options, with error 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*. Version 9 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
TSO Command Summary

Starting StarTool FDM

Issue the `STARTOOL` command at the TSO command line to start StarTool FDM. If you are already in an ISPF session when you issue the command, StarTool FDM defaults to the ISPF-mode interface (`ISPMODE`) on startup. Otherwise, the product executes in a TSO session and invokes ISPF menus externally (`XISPMODE`).

Examples
```
STARTOOL lib.cntl
TSO STARTOOL lib.cntl
```

Syntax
```
STARTOOL
    {dsn [VOLUME(vser)|VOLSET(vdef)][SHR|OLD]|
            *|
       FILE(ddname) [NUMBER(num)]}
    [ISPMODE|XISPMODE|ISPXEQ command|subcommand|MEMLIST memgroup]
```

Aliases
```
PDS, PDSE
```

Defaults

`ISPMODE` if in an ISPF session, otherwise `XISPMODE`; SHR

Required

dsn or `FILE(ddname)`

Operands

dsn	Identifies the data set name. If you fail to enter the data set name in quotes ('), your TSO PREFIX is appended to the start of the entered data set name. If you enter ***** in this position, the system assumes FILE(ISPPROF). If this is a migrated data set, StarTool FDM changes to your ISPF profile data set to avoid startup delays. If your system has password security, enter your data set password after the data set name and a slash /. The syntax is dsn/password.

VOLUME(*vser*)	Specifies the volume name on which the data set resides. If you enter a volume name, SYSALLDA is assumed as the unit name; otherwise, StarTool FDM uses the unit name from the catalog. Use this parameter if the data set is not cataloged or if the catalog entry is not to be used. To use the catalog for the **STARTOOL** command, even if VOLSET is in effect, enter VOLUME(*).
VOLSET(*vdef*)	Specifies a default volume name for data set references. If you enter VOLSET(NEWRES), subsequent **CHANGE** subcommands with a data set name and no VOLUME parameter assume VOLUME(NEWRES). Use this parameter if you want a default volume. To nullify the effect of VOLSET, enter VOLSET(*) on a subsequent **CHANGE** subcommand.
SHR	Allocate the data set with a disposition of SHR (share). Allow simultaneous use of this data set by others.
OLD	Allocate the data set with a disposition of OLD. Do not allow simultaneous use of this data set by others. The use of SHR is recommended.
FILE(*ddname*)	Identifies the DDNAME of a pre-allocated data set. Only disk data sets (including VIO) are supported. If you use the FILE keyword, then do not use dsn, SHR/OLD, VOLSET, and VOLUME. If the data set is concatenated, the data set is reallocated so you can use SHR or OLD.
NUMBER(*num*)	Specifies the concatenation number for the DDNAME allocation for the FILE keyword. *num* defaults to 1 but if *num* is larger than the number of concatenated data sets, the last data set in the sequence is used.
ISPMODE	Requests that StarTool FDM initialize in ISPMODE. This is the initial mode only; you can suspend ISPMODE and MEMLIST with a later subcommand.
XISPMODE	Requests that StarTool FDM initialize in line mode (independent of ISPF). This is the initial mode only; at a later time, you can enter an ISPMODE or MEMLIST subcommand.
MEMLIST *memgroup*	Requests that StarTool FDM initialize with a member list set to the named member group. This is the initial mode only; you can suspend MEMLIST and ISPMODE with a later subcommand.
ISPXEQ *command*	Requests that StarTool FDM initialize with the ISPMODE service (usually DDNAME, LISTA, LISTC, LISTF or LISTV) named. This is the initial mode only; you can enter other **STARTOOL** subcommands after the initial display.
subcommand	Requests that StarTool FDM perform the single subcommand and terminate. This is a special subcommand mode for performing only a single subcommand. In this mode, ISPF services are not available and StarTool FDM operates in line mode only. Also, there is no YES/NO prompting; instead, YES responses are assumed. When you invoke StarTool FDM in single subcommand mode, the return code is set to the numeric value of the first warning or error message encountered.

ISPMODE Startup Panel

The following panel displays when you start StarTool FDM from ISPF.

```
--------------------------- ISPMODE Session# 1 Log# 1 --------------- ROW 1 TO
     13 OF 13
COMMAND ===>                                                    SCROLL ===>CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER001 MEM=PDS100I StarTool/SuperEdit -- Version v.r.m

Proprietary software product of SERENA Software Inc.
LICENSED TO: your corporate name/trial offer expires ...
          your city, state, zip/agent to contact for license ...
All other rights reserved - use of this software
product by unauthorized persons is prohibited.

PDS200I DISP UNIT   RECFM LRECL BLKSIZE  ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390    FB     80  13680   1X   100      19    50 TRK      17

PDS300A ENTER OPTION -- DSN=SER07.LIB.CNTL,VOL=SER=SER001 MEM=
******************************* BOTTOM OF DATA *********************************
```

Member Names in Subcommands

Hexadecimal Member Names

Enter member names in hexadecimal for any of the subcommands that accept member names (or partial member names) as input. For these subcommands, x'd7c4e2c5' and PDSE are equivalent. Also, x'333' and x'0333' are equivalent.

Default Member Names

If you enter ***** in the member group position for an ABE, ACFCOMP, BROWSE, COMBINE, COMPDIR, COPY, DCF, DELETE, DELINK, DIRENTRY, DISASM, DUP, EDIT, EXCLUDE, FIND, FSE, HISTORY, IF, LIST, LLA, MAP, MEMBERS, MEMLIST, OUTCOPY, PGMDOC, PRINT, READOBJ, REPLACE, REPRO, REVIEW, SMPGEN, SPFEDIT, SUBLIST, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT or XREF subcommand, the **STARTOOL** command uses the last member name or member group entered for any of these subcommands.

When you specify a default member name or group by one of the above subcommands, the member name or group becomes the "current member group" or the "default member name or group" (the ALIAS, RENAME, and RESTORE subcommands also modify the default member group but do not process member groups). The current member group can be referred to by this set of subcommands until it is again redefined.

Member Groups

Subcommands that allow default member names also accept *member groups* as member names. A member group can be a single member name, a range of member names, a member name pattern, or a lists of member names or other valid member group specifications. See Appendix A, "Formatting Member Names", for more information.

1   Enter a single asterisk (*****) to refer to the currently defined member group.

2   A *member name range* is of the form `name1:name2`, which selects members based on character string ranges in their names.

3   A *pattern member name* is of the form `name1/name2`, which selects members based on character string matches in their names.

4   A *combination member name* is of the `form name1*name2`, which selects members based on the range name1:name1 and on the pattern name2 in the remainder of the member name.

5   A *placeholder* is allowed in a simple member name, a pattern member name or in a combination member name. Specify a placeholder by using a `?` or % anywhere in the member name. It acts as a single-character wildcard.

6   A *member name list* is a parenthesized list of member name specifications. It contains normal member names as well as the member group forms described above. When you use a member list, the subcommand receives control separately for each member specification in the list.

7   Enter a single equal symbol (=) to refer to the list of members in the current MEMLIST. If a MEMLIST is not active, the equal symbol is equivalent to an asterisk (the current member group).

TSO Subcommands and Functions Table

StarTool FDM supports about 100 different subcommands that you can invoke at the TSO command line. Most of these have multiple operands.

When executing a subcommand from the TSO command line, you can abbreviate subcommand names to the first few characters. You can also abbreviate any keyword operands for these subcommands. The general rule is that you abbreviate by dropping ending characters of a keyword name as long as the abbreviated keyword remains unique for the subcommand.

A summary of StarTool FDM subcommands appears below. Subsequent chapters provide detailed syntax and parameter information. Subcommands are listed in alphabetical order.

Subcommand	Short	Function	Heading	Source	Load
ABE	AB	Edits data	"ABE Subcommand"	yes	yes
ACFCOMP	ACF	Compiles source CA-ACF2 rules	"ACFCOMP Subcommand"	yes	no
ALIAS	AL	Adds an alias name for a member	"ALIAS Subcommand"	yes	yes
	A	Lists and modifies member attributes	"ATTRIB Subcommand (for source members)" and "ATTRIB Subcommand (for load members)"	yes	yes

Subcommand	Short	Function	Heading	Source	Load
BLK3380	BLK338	Optimizes 3380 disk utilization	"BLK3380 Subcommand"	yes	yes
BLK3390	BLK339	Optimizes 3390 disk utilization	"BLK3390 Subcommand"	yes	yes
BLK9345	BLK9	Optimizes 9345 disk utilization	"BLK9345 Subcommand"	yes	yes
BROWSE	B	Browses data	"BROWSE Subcommand"	yes	yes
CALC	CAL	Floating point calculator (ISPMODE only)	"CALC Function"	yes	yes
CAX	CAX	Lists active system catalogs (ISPMODE only)	"CAX Function"	yes	yes
CHANGE	C	Switches to a different data set	"CHANGE Subcommand"	yes	yes
CMDTBL	CMD	Manages ISPF command tables	"CMDTBL Function"	yes	yes
COMBINE	COMB	Joins several members together	"COMBINE Subcommand"	yes	no
COMPARE	CO	Displays data differences	"COMPARE Subcommand"	yes	yes
COMPDIR	COMPD	Compares member directory entries	"COMPDIR Subcommand"	yes	yes
COMPRESS	COMPR	Compresses data sets	"COMPRESS Subcommand"	yes	yes
CONDEND	COND	Conditionally terminates StarTool FDM	"CONDEND Subcommand"	yes	yes
CONTROL	CON	Controls StarTool FDM global options	"CONTROL Subcommand"	yes	yes
COPY	COP	Copies data selectively	"COPY Subcommand"	yes	yes
CREATE	CR	Creates a data set based on the current data set	"CREATE Subcommand"	yes	yes
CSECTS	CS	Maps module CSECTs (ISPMODE only)	"CSECTS Function"	no	yes
DCF	DC	Scripts a data set	"DCF Subcommand"	yes	no

Subcommand	Short	Function	Heading	Source	Load
DDNAME	DD	Lists TSO session allocations (ISPMODE only)	"DDNAME Function"	yes	yes
DECODE	DEC	Decrypts a member	"DECODE Subcommand"	yes	no
DELETE	DEL	Deletes members	"DELETE Subcommand"	yes	yes
DELINK	DELI	Produces object code from a load module	"DELINK Subcommand"	no	yes
DIRENTRY	DIR	Interprets member directory entries	"DIRENTRY Subcommand"	yes	yes
DISASM	DISA	Disassembles load modules	"DISASM Subcommand"	no	yes
DISPLAY	D	Lists member names from the directory	"DISPLAY Subcommand"	yes	yes
DSAT	DSA	Displays data set attributes	"DSAT Subcommand"	yes	yes
DSNAME	DS	Displays data set allocation information	"DSNAME Subcommand"	yes	yes
DUP	DU	Copies data selectively	"DUP Subcommand"	yes	yes
DVOL	DV	Displays volume attributes and statistics	"DVOL Subcommand"	yes	yes
EDIT	E	Edits data	"EDIT Subcommand"	yes	no
EDREC	EDR	Invokes ISPF edit recovery	"EDREC Subcommand"	yes	no
ENCODE	ENC	Encrypts a member	"ENCODE Subcommand"	yes	no
END	EN	Terminates the current function of StarTool FDM	"END Subcommand"	yes	yes
EXCLUDE	EXC	Drops members from MEMLIST	"EXCLUDE Subcommand"	yes	yes
EXEC	EX	Executes a CLIST with subcommands	"EXEC Subcommand"	yes	yes
FILTER	FIL	Filters data sets for LISTC/LISTF and MASK	"FILTER Function"	yes	yes

Subcommand	Short	Function	Heading	Source	Load
FIND	FI	Searches for a string	"FIND Subcommand"	yes	yes
FINDMOD	FINDM	Locates system modules	"FINDMOD Subcommand"	yes	yes
FIXPDS	FIX	Modifies a data set	"FIXPDS Subcommand"	yes	yes
FSE	FS	Edits data	"FSE Subcommand"	yes	no
GO	GO	Switches StarTool FDM sessions (ISPMODE only)	"GO Function"	yes	yes
HELP	H	Provides information on using StarTool FDM	"HELP Subcommand"	yes	yes
HEX	HEX	Hexadecimal calculator (ISPMODE only)	"HEX Function"	yes	yes
HISTORY	HI	Displays CSECT IDR data from a module	"HISTORY Subcommand"	no	yes
IDCAMS	IDC	Invokes an IDCAMS command	"IDCAMS Subcommand"	no	no
IF	IF	Searches for members with desired attributes	"IF Subcommand"	yes	yes
ISPF	ISPF	Stacks an ISPF session	"ISPF Subcommand"	yes	yes
ISPMODE	ISPM	Switches StarTool FDM to an ISPF display mode	"ISPMODE Function"	yes	yes
ISPXEQ	ISPX	Invokes StarTool FDM with an ISPMODE option	"ISPXEQ Function"	yes	yes
LIST	LI	Displays data	"LIST Subcommand"	yes	yes
LISTA	LA	Lists TSO session allocations (ISPMODE only)	"LISTA Function"	yes	yes
LISTC	LC	Lists data sets from a catalog (ISPMODE only)	"LISTC Function"	yes	yes

Subcommand	Short	Function	Heading	Source	Load
LISTF	LF	Lists data sets from a volume (ISPMODE only)	"LISTF Function"	yes	yes
LISTGRP	LISTG	Displays the member group names	"LISTGRP Subcommand"	yes	yes
LISTV	LV	Lists disk volumes (ISPMODE only)	"LISTV Function"	yes	yes
LLA	LLA	Refreshes LLA directory entries	"LLA Subcommand" and "LLA and Deleted Members"	yes	yes
LOG	LO	Selects one of 9 session logs (ISPMODE only)	"LOG Function"	yes	yes
MAP	MA	Displays the CSECT structure of a module	"MAP Subcommand"	no	yes
MASK	MAS	Front-end LISTC and LISTF (ISPMODE only)	"MASK Function"	yes	yes
MEMBERS	ME	Lists member names in a member group	"MEMBERS Subcommand"	yes	yes
MEMLIST	ML	Builds a member list table (ISPMODE only)	"MEMLIST Function" and "MEMLIST and Deleted Members"	yes	yes
NUCMAP	NUC	Maps nucleus (ISPMODE only)	"NUCMAP Function"	yes	yes
OPTIONS	O	Provides a directory of subcommands	"OPTIONS Subcommand"	yes	yes
OUTCOPY	OUTC	Formats utility control statements	"OUTCOPY Subcommand"	yes	yes
PATTERN	P	Displays member names from the directory	"PATTERN Subcommand"	yes	yes
PBROWSE	PB	Browses data (ISPMODE only)	"PBROWSE Function"	yes	yes
PEDIT	PE	Edits data (ISPMODE only)	"PEDIT Function"	yes	yes
PVIEW	PV	Edits data in VIEW mode (ISPMODE only)	"PVIEW Function"	yes	yes

Subcommand	Short	Function	Heading	Source	Load
PGMDOC	PGM	Lists module descriptions	"PGMDOC Subcommand"	no	yes
PRINT	PR	Prints a hardcopy listing	"PRINT Subcommand"	yes	yes
PROFMAN	PRO	Manages saved tables (ISPMODE only)	"PROFMAN Function"	yes	yes
QUIT	Q	Terminates StarTool FDM	"QUIT Subcommand"	yes	yes
READOBJ	READ	Disassembles object code	"READOBJ Subcommand"	yes	no
RECALL	R	Displays the previous subcommand	"RECALL Subcommand"	yes	yes
RENAME	REN	Changes a member's name	"RENAME Subcommand"	yes	yes
REPLACE	REPL	Changes member contents	"REPLACE Subcommand"	yes	yes
REPRO	REPR	Rebuilds or creates members	"REPRO Subcommand"	yes	yes
RESTORE	RES	Resurrects deleted members	"RESTORE Subcommand"	yes	yes
REVIEW	REV	Browses a member without ISPF	"REVIEW Subcommand"	yes	yes
SEPARATE	SEP	Splits into members	"SEPARATE Subcommand"	yes	no
SMPGEN	SM	Generates SMP/E sysmods	"SMPGEN Subcommand"	yes	yes
SUBLIST	SUBL	Creates a member subset	"SUBLIST Subcommand"	yes	yes
SUBMIT	SUB	Submits data for background processing	"SUBMIT Subcommand"	yes	no
STATUS	ST	Displays the status of ISPMODE functions	"STATUS Function"	yes	yes
SVCMAP	SVC	Investigates SVC routines	"SVCMAP Subcommand"	yes	yes
TSO	T	Invokes a TSO command processor or CLIST	"TSO Subcommand"	yes	yes

Subcommand	Short	Function	Heading	Source	Load
TSOEDIT	TSOE	Edits data	"TSOEDIT Subcommand"	yes	no
TSOLIST	TSOL	Lists data	"TSOLIST Subcommand"	yes	yes
USAGE	U	Displays data set statistics	"USAGE Subcommand"	yes	yes
VERIFY	V	Validates members or a data set	"VERIFY Subcommand"	yes	yes
VMAP	VM	Displays a disk volume map (ISPMODE only)	"VMAP Function"	yes	yes
VPRINT	VP	Prints on a VTAM printer	"VPRINT Subcommand"	yes	yes
VSAM		Invokes VSAM Services	"VSAM Subcommand"		
VTOC	VT	Displays data sets on disk volumes	"VTOC Subcommand"	yes	yes
VUSE	VU	Displays disk usage and statistics	"VUSE Subcommand"	yes	yes
WHOHAS	WH	Checks allocation status of a data set	"WHOHAS Subcommand"	yes	yes
WORKPAD	W	Stores or executes commands (ISPMODE only)	"WORKPAD Function"	yes	yes
XREF	XR	Displays intra-module references	"XREF Subcommand"	no	yes
ZAP	Z	Modifies load module (ISPMODE only)	"ZAP Function"	no	yes

Chapter 2
ISPF Interface Commands

This section lists all interactive StarTool FDM commands in alphabetical order. Defaults are underlined and the shortest valid abbreviation for a command name is shown before the [brackets]. For example, a command shown as COMM[ENT] could be specified as COMM, COMME, COMMEN, or COMMENT.

Common Commands

This section documents ISPMODE-only commands that you can enter in any ISPMODE function. <Xref>"All Commands" documents all ISPMODE only commands that can be entered (including commands specific to a single function).

The following commands are supported by StarTool FDM anywhere in ISPMODE:

Command	Description	
*	Merges current member group members into the MEMLIST table. `*cm ERR- t="Command,*">`	
?	Provides extended help on up to five warning or error messages from the last subcommand.	
/*	Allows any following information to be entered as comments. `*cm ERR- t="Command,/*">`	
ALIASCHK	Alias check. Checks aliases and adds all associated members to the MEMLIST display.	
ALL	MEMLIST all. Adds all members in the data set to the MEMLIST display.	
ALLOC[ERR]	Goes to a tutorial explaining dynamic allocation error codes.	
ALTC[MD]	Executes a subcommand (previously saved by SETALT) on the current data set.	
ALT[ERNAT]	Displays an alternate view of the current table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
BLK[SIZE]	Provides information on disk track utilization for optimal blocking.	
CAN[CEL]	Cancels pending line commands for the function specified.	
	Syntax	`CAN CSE|CAX|LA|LF|LV|ML|W|ALL`
CI[SIZE]	Provides information on optimal control interval sizes for VSAM data sets.	
COMM[ENT]	Allows following information to be entered as comments.	
DEF[INE]	Provides the equivalent of IDCAMS; DEFINE	

`DR[OP]`	Terminates an active GO session by number.	
	Syntax	`DROP [*/n] [PROMPT]`
	(where *n* defaults to the current GO session)	
`DUA[L]`	Displays a double or triple line view of the current table.	
`EDITL[OG]` `(or EL[OG])`	Enters an edit session on the output from the last subcommand.	
`EDITT[BL]` `(or ET[BL])`	Enters an edit session on data from the current function table.	
`EQ[UATE]`	SUBLIST =. Resets the current member group to the members in the current MEMLIST.	
`EXPA[ND]`	Prompts for FIXPDS operands that add directory blocks and disk space.	
`GR[OUP]`	Provides a prompt panel for specifying a member group followed by a prompt for a subcommand.	
`IN[DEX]`	Goes to the StarTool FDM tutorial index.	
`LAS[TCMDS]`	Presents the last 32 primary commands for selection by number for modification and reuse.	
`LE[FT]`	Rotates through alternate views of the current data. PF keys 10 and 22 are normally set to LEFT.	
`MC[OPY]`	Interfaces with the extended copy feature using a default data set name as set in MODEL.	
`M[ENU]`	Provides menu system command and operand entry assistance. Also use MENU to switch between the StarTool FDM primary option panel and the primary panels for licensed options. You can chain items as in M.3.3 (or its equivalent, 3.3). You can invoke entry assist panels for most subcommands as in M.FIND (or its equivalent, FIND).	
	Syntax	`MENU [ON|OFF]`
`MODEL`	Provides a prompt panel (with initial data set values) for the CREATE or IDCAMS subcommand.	
`MON[TH]`	Merges members updated or created this month into the MEMLIST.	
`NOR[MAL]`	Displays the default view of the current table.	
`O[PTIONS]`	Provides primary command selection for the current function and operand syntax assistance.	
`OUT[PUT]`	Outputs the current function table to print or to a data set.	
	Syntax	`OUTPUT [=c / F(ddname)]`
`PAN[EL]`	Selects a panel using ISPF display services (normally used for panel testing).	
	Syntax	`PANEL member`
`PEND`	Checks for pending line commands and selects the next one. If no line commands are pending, the command is ignored or the primary MENU is selected if MENU mode is enabled.	
`PRIM[ER]`	Goes to a StarTool FDM introductory tutorial.	

R[ECALL] (or RC)	Displays the last StarTool FDM subcommand for modification and rentry. RECALL also retrieves subcommands from the log.
REL[EASE]	Prompts for FIXPDS operands that release excess disk space.
RF[IND]	Finds a string (repeat find). In the log, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location. PF keys 5 and 17 are normally set to RFIND.
RI[GHT]	Rotates through alternate views of the current data. PF keys 11 and 23 are normally set to RIGHT.
SET	Prompts for one of the following SET services.
SETA[LL]	Controls StarTool FDM combined defaults
SETALT	Recalls and saves a subcommand for repeated use later by ALTCMD.
SETC[OLOR]	Controls screen colors and highlighting.
SETD[SN]	Sets data set controls.
SETK[EYS]	Controls PF keys for StarTool FDM panels (for ISPF 4.x users, the KEYS command can be used for the same effect as SETKEYS).
SETP[ANEL]	Controls optional panels and panel defaults.
SETSEEK	Specifies the default member name for the SEEK line command.
SETSEL	Specifies alias names for S (or SELECT) by function.
SETU[SER]	Controls dynamic primary commands. Specify command names and their actions.
SUS[PEND]	Terminates ISPMODE temporarily and enters line mode processing. Restart ISPMODE with the same log table by entering an ISPMODE, ISPXEQ, or MEMLIST subcommand. While ISPMODE is suspended, IND$FILE is supported for PC SEND and RECEIVE.
TAG	Forms a member group containing only members marked with *TAG* in the DATA/MSG field for MEMLIST. Otherwise, applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
TOD[AY]	Merges members updated or created today into the MEMLIST.
TRANS	Provides a translation service to convert a hex, decimal, or EBCDIC character to the other forms.
	Syntax `TRANS 1-to-2-hex|1-to-3-decimal|char` `                 [HEX|NUMBER|CHAR]`
TRAP	Interfaces with TSO commands to capture their output lines in the log.
	Syntax `TRAP tsocommand [anyoperands]`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. Several of these panels also allow dynamic primary commands with which you specify command names and their corresponding actions.
WE[EK]	Merges members updated or created this week into the MEMLIST.
X	Clears the current table relative to the cursor position.

	Syntax	`X [ABOVE|BELOW|ALL]`
`XA[LL]`	Clears the current table. This is equivalent to X ALL	
`XC[OPY]`	Interfaces with the extended copy feature.	
`XM[IT]`	Transmits the data set to another user.	

All Commands

This section documents all ISPMODE-only commands that can be entered (including commands specific to a single function).

Global commands are **STARTOOL** subcommands that apply to all data sets in a DDNAME/ LISTA, LISTC/LISTF or WORKPAD table. A command named GLOBAL is a general global command. Follow GLOBAL with any **STARTOOL** subcommand. The subcommand is issued against each data set in the table.

The FIND, REPLACE, COMPRESS, MODEL, SEEK and WHOHAS commands are also global for these tables. In LISTC/LISTF tables, the FIND, REPLACE, COMPRESS and SEEK commands skip over non-partitioned data sets. This happens because the FIND and REPLACE subcommands use different subcommand syntax for partitioned data sets and non-partitioned data sets. If you want to search or update non-partitioned data sets, type GLOBAL followed by FIND or REPLACE and its operands.

The following commands are supported by StarTool FDM in ISPMODE. Several of these commands are supported only in a single function:

`*`	Merges current member group members into the MEMLIST table. `*cm ERR- t="Command,*">`
`?`	Provides extended help on up to five warning or error messages from the last subcommand.
`/*`	Allows any following information to be entered as comments. `*cm ERR- t="Command,/*">`
`ADD`	In WORKPAD, adds 5 blank table lines to the end of the WORKPAD table. In LISTV, adds data sets to the LISTF table for all volumes in the LISTV table without prompting.
`ALIASCHK`	Alias check. Checks aliases and adds all associated members to the MEMLIST display.
`ALL`	MEMLISTs all. Adds all members in the data set to the MEMLIST display.
`ALLOC[ERR]`	Goes to a tutorial explaining dynamic allocation error codes.
`ALTC[MD]`	Executes a subcommand (previously saved by SETALT) on the current data set.
`ALT[ERNAT]`	Displays an alternate view of the current table
`APP[LY]`	Applies the specified line command to all table entries and executes each entry.
	Syntax `APPLY linecmd`

BA[SE]	In ZAP, resets the base address for this CSECT (like an AMASPZAP BASE statement).	
	Syntax	`BASE hexaddress`
	(1 to 6 hexadecimal digits)	
BAT[CHJCL]	In DDNAME, LISTA, LOG and MEMLIST builds batch JCL from information in the table.	
BLK[SIZE]	Provides information on disk track utilization for optimal blocking.	
BOU[NDS]	In PEDIT, specifies default find column limits. BOUNDS OFF and BOUNDS with no operand turn off bounds processing. BOUNDS LIST shows current boundaries.	
	Syntax	`BOUNDS nn mm| OFF | LIST`
BU[ILD]	In ZAP, outputs AMASPZAP format output without changing the data set. Note that AMASPZAP backout controls are also formatted as comments.	
CAN[CEL]	<u>In LOG</u> after a reply required for RESTORE, specifies that StarTool FDM terminate RESTORE processing. A YES response allows the restoration of the current deleted member and a NO response skips the current deleted member for RESTORE processing. Otherwise, cancels pending line commands for the function specified.	
	Syntax	`CAN CSE|CAX|LA|LF|LV|ML|W|ALL`
CAN[CEL]	In PEDIT, terminates the edit session with no data updates.	
CB	In PEDIT, prompts for a COBOL, Assembler or PL/I copybook name to overlay define edit records.	

	Syntax
	`CB [ON]|[OFF]|[NEW]|[FIRST]|[NEXT]|[PREV]`
	    `[ON]   -- default, switches to copybook mode`
	    `[OFF]  -- reverts to the previous PEDIT display`
	    `[NEW]  -- prompts for a new copybook member`
	    `[FIRST]-- switches to the first copybook with matching values`
	    `[NEXT] -- switches to the next base copybook that matches`
	    `[PREV] -- switches to the previous base copybook that matches`	

CBMAP	In PEDIT, prompts for a COBOL, Assembler or PL/I copybook name to display independently of edit records. CBMAP also sets the default copybook data set and member for the CB line command.	
C[HANGE]	In PEDIT, updates a character string or a hexadecimal string.	

	Syntax
	`CHANGE anystring newstring [nn mm] [ASIS] [ALL]`
	        `[FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`	

	(where *nn* and *mm* are column numbers)	
	The two strings must be the same length.	
CI[SIZE]	Provides information on optimal control interval sizes for VSAM data sets.	
CLOSE	In PEDIT, same as END, updates changed records and terminates PEDIT.	
COLS	In LOG, provides a column ruler in the log for determining column numbers.	

CO[LS]	In PBROWSE, displays a columns line on the first line of the data area. The columns line remain at the top of the data display. It is useful in identifying columns to be used with the FIND command.	
	Syntax	COLS [ON|OFF]
COMM[ENT]	Allows any following information to be entered as comments.	
COMPR[ESS]	Global command in LISTA/DDNAME, LISTC/LISTF, and WORKPAD, changes to each data set in the table and issues a COMPRESS subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.	
	Syntax	COMPRESS [LIST|NOLIST|SUMMARY]
CONT[INUE]	In LOG after a checkpoint, specifies that StarTool FDM continue the current interrupted process until the next checkpoint interval as specified by SETALL. Use END to terminate the process.	
COPYBOOK	In PEDIT, prompts for a COBOL, Assembler or PL/I copybook name to overlay define edit records. See CB for keyword details.	
	Syntax COPYBOOK [ON]|[OFF]|[NEW]|[FIRST]|[NEXT]|[PREV]	
CREATE	In PEDIT, creates a new partitioned data set member from marked copy or move data lines in an edit session.	
	Syntax	CREATE [member]
	(if a member name is not specified, extended CREATE is used)	
CUT	In PEDIT, places marked edit sessions lines into the StarTool FDM clipboard. In a later PEDIT or ISPF edit session, use PASTE to place these lines back into a session.	

	Syntax
	 CUT [mm] [ADD|REPL] [DEFAULTS]
	    mm -- two character table name
	    ADD -- add to the current clipboard
	    REPL -- replace clipboard contents
	    DEFAULTS -- display a panel to set defaults	

DEF[INE]	Provides the equivalent of IDCAMS; DEFINE.	
DIS[PLAY]	In PEDIT, displays operands and defaults for COPYBOOK, SCREEN, VERTICAL, SHOW, and HIDE.	
DR[OP]	Terminates an active GO session by number.	
	Syntax	DROP [*/n] [PROMPT]
	(where *n* defaults to the current GO session)	
DUA[L]	Displays a double or triple line view of the current table.	
EDITL[OG] (or EL[OG])	Enters an edit session on the output from the last subcommand.	
EDITT[BL] (or ET[BL])	Enters an edit session on data from the current function table.	
END	In PEDIT, same as CLOSE, updates changed records and terminates PEDIT.	
EQ[UATE]	SUBLIST =. Resets the current member group to the members in the current MEMLIST.	

ER[ASE]	In LISTC/LISTF and WORKPAD, deletes the current table in memory and disk.
EXPA[ND]	Prompts for FIXPDS operands that add directory blocks and disk space.
EXPR[ESS]	In CAX, CSECTS, LISTA/DDNAME, LISTC/LISTF, LISTV, MEMLIST, NUCMAP, WORKPAD, and ZAP executes all entered line commands without pauses between individual commands.
F	Finds a string. In LOG, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location.

	Syntax	`F anystring  [ASIS]` `        [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`

FI[ND]	Global command in LISTA/DDNAME, LISTC/LISTF, and WORKPAD, changes to each data set in the table and issues a FIND subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped. In other tables, do not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.

	Syntax	`FIND memgroup 'anystring'`
		`        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
		`        [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`

F[IND]	In PEDIT, finds a string and positions the display start location.

	Syntax

	`FIND anystring [nn mm] [ASIS] [ALL]`
	`        [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	`        [CHECK] -for copybook, searches for invalid packed decimal`
	`        [NAME] -for copybook, positions to a copybook variable name`
	`        [SAME] - for copybook, searches only the current record`

	(where *nn* and *mm* are column numbers)

F[IND]	In PBROWSE, finds a string and positions the display start location.

	Syntax	`FIND anystring [nn mm] [ASIS]` `        [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`

	(where *nn* and *mm* are column numbers)
GL[OBAL]	Global command in LISTA/DDNAME, LISTC/LISTF, and WORKPAD changes to each data set in the table and issues any StarTool FDM subcommand.

	Syntax	`GLOBAL anysubcommand [anyoperands]`

GR[OUP]	Provides a prompt panel for specifying a member group followed by a prompt for a subcommand.
HEX	In PBROWSE and PEDIT, sets HEX display mode on or off.

	Syntax	`HEX [ON|OFF]`

HIDE	In PEDIT, for copybook display or VERTICAL display, excludes named or numbered elements from the current display.

	Syntax
	` HIDE {ALL|varnam|gennam|n1 [TO] n2} GROUP`
	    ALL -- all variables are to be hidden
	    varnam -- exact variable name from the copybook
	    gennam -- partial variable name like FORM-DETAIL*
	    n1 -- entry number for a variable from SCREEN NUMBER
	    TO -- used with n2 to indicate a range
	    n2 -- a second entry number to specify a range end with TO
	    GROUP -- hides all variables from a given copybook level

`ID`	In LISTC/LISTF and WORKPAD, changes the table ID name.
	Syntax `ID name`
	(where *name* is a 1- to 6-character alphanumeric name).
`IDR[DATA]`	In ZAP, supplies the IDRDATA for the current ZAP.
	Syntax `IDRDATA idrname`
`IN[DEX]`	Goes to the StarTool FDM tutorial index.
`INS[ERT]`	In WORKPAD, adds blank WORKPAD table lines.
	Syntax `INSERT [nn] [TOP|BOTTOM|HERE]`
	(where *nn* defaults to 5)
`KEY`	In PEDIT, positions directly to the record with the corresponding record key.
	Syntax `KEY 'record key'	x'hexkey'`
`LAS[TCMDS]`	Presents the last 32 primary commands for selection by number, for modification and reuse.
`LE[FT]`	Rotates through alternate views of the current data. PF keys 10 and 22 are normally set to LEFT.
`LISTVT[OC]`	In LISTV, adds data sets to the LISTF table for all volumes in the LISTV table after a prompt.
`LO[AD]`	In LISTC/LISTF and WORKPAD, loads data set names from operating system control blocks.
	Syntax `LOAD APFLST|LNKLST|LPALST  [RESET]`
`L[OCATE]`	In CAX, CSECTS, LISTC/LISTF, LISTV, MEMLIST, NUCMAP, and ZAP positions to a data line in sorted tables by searching the current sorted column for the specified data. Type LOCATE without operands to go to a prompting panel.
	Syntax `LOCATE datavalue`
`L[OCATE]`	In PBROWSE, positions directly to the specified record number.
	Syntax `LOCATE recnumber`
`L[OCATE]`	In PEDIT, positions directly to a record number or duplicate record (by key).
	Syntax `LOCATE recnumber [DUP] [FIRST]`
`MC[OPY]`	Interfaces with the extended copy feature using a default data set name as set in MODEL.

M[ENU]	Provides menu system command and operand entry assistance. Also use MENU to switch between the StarTool FDM primary option panel and the primary panels for other licensed options. Items can be chained as in M.3.3 (or its equivalent, 3.3). In addition, entry assist panels for most subcommands can be invoked as in M.FIND (or its equivalent, FIND).	
	Syntax	`MENU [ON|OFF]`
MER[GE]	In MEMLIST, LISTC/LISTF and WORKPAD, merges data from a different table into the current table.	
	Syntax	`MERGE name [GROUP|SAMPLE] [RESET]`
	(where *name* is 1 to 6 alphanumeric characters)	
	Syntax	`[TOP|BOTTOM|HERE]`
	(for WORKPAD only)	
MINE	In MEMLIST, builds a member list of members with ISPF statistics that were saved by the current USERID.	
MODE	In WORKPAD, specifies whether or not modified WORKPAD lines are to be executed automatically.	
	Syntax	`MODE [EXEC|NOEXEC]`
MODEL	<u>global command</u> in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and issues a MODEL command. Otherwise, provides an entry assist panel (with initial data set values) for the CREATE or IDCAMS subcommand.	
MON[TH]	Merges members updated or created this month into the MEMLIST.	
NEXT	In PEDIT, positions to a following logical record. Use SET to change the default number of records to move forward if no operands are entered. NEXT 1 SET is the default. SET changes the default and causes a new record to display.	
	Syntax	`NEXT [num SET]|[MAX]`
NO	In LOG after a required reply, specifies that StarTool FDM not complete the current action. A YES response allows the process to continue. For a RESTORE subcommand, type CAN to terminate RESTORE processing.	
NOR[MAL]	Displays the default view of the current table.	
NOS[AVE]	In LISTC/LISTF and WORKPAD, specifies that the current table not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF or WORKPAD.	
NOZ[AP]	In ZAP, exits the ZAP function without changing additional data on disk.	
O[PTIONS]	Provides primary command selection for the current function and operand syntax assistance.	
OFF[SET]	In ZAP, resets the beginning offset for CSECT displays. Lower offsets are not displayed.	
	Syntax	`OFFSET hexaddress`
	(1 to 6 hexadecimal digits)	
OUT[PUT]	Outputs the current function table to print or to a data set.	

	Syntax	`OUTPUT [=c	F(ddname)]`
PAN[EL]	Selects a panel using ISPF display services (normally used for panel testing).	
	Syntax	`PANEL member`
PASTE	In PEDIT, places lines from the StarTool FDM clipboard into the active edit session. Use the PEDIT or ISPF edit CUT macro to put lines into the clipboard.	

	Syntax
	`PASTE [mm] [ZERO|KEEP] [DEFAULTS]`
	  mm -- Two character table name
	  ZERO -- Insert clipboard contents and clear clipboard
	  KEEP -- Insert clipboard contents and keep clipboard
	  DEFAULTS -- Display a panel to set defaults	

PAD	In PBROWSE and PEDIT, changes the padding character and specifies the number of pad characters desired.	
	Syntax	`PAD character|'character'|x'hex'  [column]`
	Example: PAD a 45	
PEND	Checks for pending line commands and selects the next one. If no line commands are pending, the command is ignored or the primary MENU is selected if MENU mode is enabled.	
PREV[IOUS]	For PEDIT copybook, positions to a previous logical record. Use SET to change the default number of records to move backward if no operands are entered. PREV 1 SET is the default. SET changes the default and causes a new record to display.	
	Syntax	`PREVIOUS [num SET]|[MAX]`
PRIM[ER]	Goes to a StarTool FDM introductory tutorial.	
RCH[ANGE]	In LOG, captures the command or data set name under the cursor in the log for modification and/or reuse. PF keys 6 and 18 are set to RCHANGE.	
RCH[ANGE]	In PEDIT, changes a string (repeat change) and positions the display start location. PF keys 6 and 18 are set to RCHANGE.	
R[ECALL] (or RC)	Displays the last StarTool FDM subcommand for modification and reentry. RECALL can also retrieve subcommands from the log.	
REF[RESH]	Global command in LISTC/LISTF and LISTV, updates information and status for all lines in the table.	
REL[EASE]	Prompts for FIXPDS operands that release excess disk space.	
REM[OVE]	In CAX, CSECTS, LISTA/DDNAME, LISTC/LISTF, LISTV, MEMLIST, NUCMAP, and WORKPAD, trims a dialog table based on a string match.	
	Syntax	`REMOVE anystring [columnname|ALL]` `            [PREFIX|SUFFIX|WORD] [NOT]`
REPL[ACE]	Global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD changes to each data set in the table and issues a REPLACE subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped. In other tables, do not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.	

	Syntax
	``
	 REPLACE memgroup 'fromstring' 'tostring'
	          [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]
	          [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]
	          [WRITE|NOWRITE]
	``	

REPL[ACE]	In PEDIT, replaces a data set or member from marked copy or moves data lines in an edit session.	
	Syntax	`REPLACE  [member]`
b	(if a member name is not specified, extended REPLACE is used)	
RESET	In PEDIT, clears all pending line commands without letting them execute.	
RESET	In DDNAME, LISTA, LISTC, LISTF, and WORKPAD, often used with data set tagging to clear the DATA/MSG field in all table entries.	
RF[IND]	In PBROWSE and PEDIT, finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are set to RFIND.	
RF[IND]	Finds a string (repeat find). In the log, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location. PF keys 5 and 17 are set to RFIND.	
RI[GHT]	Rotates through alternate views of the current data. PF keys 11 and 23 are normally set to RIGHT.	
SAM[PLE]	In WORKPAD, merges table number 99 from the installation table library (usually ISPTLIB) into the current WORKPAD table. Table 99 is an index to other sample WORKPAD tables. Include the additional sample tables by using the MERGE command.	
SAVE	In MEMLIST, LISTC/LISTF and WORKPAD, creates a permanent table for use in a later StarTool FDM session.	
	Syntax	`SAVE [name] [REPLACE|NOREPL]`
	(where *name* is 1- to 6-alphanumeric characters)	
SCREEN	For PEDIT copybook only, changes the display to show different types of data in the data type field and the displayed data field as shown in the table below.	
	Syntax ``` SCREEN [TYPE|PICTURE|OFFSET|COLUMN|NUMBER] [DATA|NODATA] ``` This table contains the copybook screen part, the header, and an example: | COBOL level and name	LEVEL AND NAME	02 SAMPLE-FIRST
data type=TYPE	LENGTH TYP	1 PAC
data type=PICTURE	PICTURE	S99V9
data type=OFFSET	OFFSET	62
data type=COLUMN	COLUMN	63
data type=NUMBER	NUMBER	4
displayed data=DATA	DATA	2.1
displayed data=NODATA	OFFSET	62		
SEEK	Global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD changes to each data set in the table and looks for a member. In LISTC/LISTF, non-partitioned data sets are skipped.	
	Syntax	`SEEK member`

S[ELECT]	In MEMLIST and LOG, BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.	
	Syntax	`SELECT member`
SET	Prompts for one of the following SET services.	
SETA[LL]	Controls StarTool FDM combined defaults.	
SETALT	Recalls and saves a subcommand for repeated use later by ALTCMD.	
SETC[OLOR]	Controls screen colors and highlighting.	
SETD[SN]	Sets data set controls.	
SETK[EYS]	Controls PF keys for StarTool FDM panels (for ISPF 4.x users, you can use the KEYS command for the same effect as SETKEYS).	
SETP[ANEL]	Controls optional panels and panel defaults.	
SETSEEK	The default member name for the SEEK line command.	
SETSEL	Alias names for S (or SELECT) by function.	
SETU[SER]	Controls dynamic primary commands. You specify command names and actions.	
SHOW	In PEDIT, for copybook display or VERTICAL display, includes named or numbered elements in the current display.	

	Syntax
	 SHOW {ALL|varnam|gennam|n1 [TO] n2} GROUP ONLY
	     ALL -- All variables are to be shown
	     varnam -- Exact variable name from the copybook
	     gennam -- Partial variable name like FORM-DETAIL*
	     n1 -- Entry number for a variable from SCREEN NUMBER
	     TO -- Used with n2 to indicate a range
	     n2 -- A second entry number to specify a range end with TO
	     GROUP -- Selects all variables in a given copybook level
	     ONLY -- Hides all variables except those identified	

SO[RT]	In CAX, CSECTS, LISTC/LISTF, LISTV, MEMLIST, or NUCMAP, sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order. If you enter an invalid sort command such as SORT xx, a prompt shows the valid sort fields for that function.	
	Syntax	`SORT [name] [ASCEND|DESCEND]`
SORT	In PEDIT for a VSAM KSDS data set, sorts records in ascending order by key.	
	Syntax	`SORT`
SPA[CE]	In LISTV, updates volume space statistics for each volume in the table.	
STATS	In LISTV, updates volume attributes for each volume in the table.	
SUS[PEND]	Terminates ISPMODE temporarily and enters line mode processing. ISPMODE can be restarted with the same log table by entering an ISPMODE, ISPXEQ or MEMLIST subcommand. While ISPMODE is suspended, IND$FILE is supported for PC SEND and RECEIVE.	

TAG	In MEMLIST, forms a member group containing only members marked with *TAG* in the DATA/MSG field. *Otherwise*, applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
TOD[AY]	Merges members updated or created today into the MEMLIST.	
TRANS	Provides a translation service to convert a hex, decimal or EBCDIC characters.	
	Syntax	`TRANS {1-to-2-hex|1-to-3-decimal|char}` `       [HEX|NUMBER|CHAR]`
TRAP	Interfaces with TSO commands to capture their output lines in the log.	
	Syntax	`TRAP tsocommand [anyoperands]`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. Several of these panels also allow dynamic primary commands with which you specify command names and their corresponding actions.	
VERT[ICAL]	In PEDIT, prompts for a COBOL, Assembler or PL/I copybook name to display records in a tabular copybook format Copybook elements to display can be selected or excluded with the SHOW and HIDE commands.	
	Syntax VERTICAL [ON|OFF|NEW] [DEFAULT] [NAME|NONAME] [PICT|NOPICT] [COL|NOCOL] [NUM|NONUM] [ONLY] [LEV1|NOLEV1] [LEV88|NOLEV88] ON -- Turns on VERTICAL with the current copybook OFF -- Turns off VERTICAL mode NEW -- Requests a new copybook for VERTICAL display DEFAULT -- Sets header to NAME, NOPICT, TYPE, COL, NUM NAME|NONAME -- Controls display of variable name PICT|NOPICT -- Controls display of variable picture TYPE|NOTYPE -- Controls display of variable type COL|NOCOL -- Controls display of variable start column NUM|NONUM -- Controls display of variable reference number ONLY -- Display only selected headers; otherwise, additive LEV1|NOLEV1 -- Controls if the LEVEL 1 variable is displayed LEV88|NOLEV88 -- Controls if LEVEL 88 items are displayed		
WE[EK]	Merges members updated or created this week into the MEMLIST.	
WHO[HAS]	Global command in LISTC/LISTF and WORKPAD, issues a WHOHAS subcommand for each data set in the table.	
X	Clears the current table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the current table; this is equivalent to X ALL	
XC[OPY]	Interfaces with the extended copy feature.	
XM[IT]	Transmits the data set to another user.	

Y[ES]	In LOG, after a required reply, specifies that StarTool FDM continue the current action. A NO response stops the current action. For a RESTORE subcommand, type CAN to terminate RESTORE processing.
ZAP	In ZAP, updates data lines in *CHA (changed) status on disk permanently.

Chapter 3

Commands — A to C

Subcommands and Functions

Subcommands are requests for operations that are supported in all StarTool FDM environments.

A function command is a request for an ISPMODE operation. Most functions support their own set of commands. Function commands control ISPF tables that are available throughout a StarTool FDM session in parallel mode.

While operating in a StarTool FDM session, commands are ISPMODE-only requests for an operation.

Enter subcommands, function commands and commands, as primary commands; that is, from the top command entry line of a panel. In most function tables, enter line commands in the CMD column. A line command is a request for an ISPMODE-only operation that affects only the table entry on which it is entered.

In each function that supports line commands, = and X are available as line commands. Use an = line command to repeat the previously entered line command. Use X line commands to drop table entries selectively. Follow these line commands by a number (for example, =3 or X3) to operate on multiple table lines.

Use block line commands to operate on multiple table entries. For example, use the XX block line command to mark table elements to drop from a table. XX operates on a range of entries. The first table element with an XX command is considered the first entry in the range. All following elements are dropped up to a paired entry with another XX block line command. XX and == block line commands are available in each function that supports line commands.

Commands, line commands, and block line commands are documented in this section but only under the function in which they are supported. For a discussion of all ISPMODE commands, see "All Commands".

Descriptions of commands, subcommands, and functions A through C follow.

ABE Subcommand

The ABE subcommand edits data; use the TSO ABE command. ABE is an acronym for A Better Editor. Add any operand after the member name.

Example

```
ABE    mema:memb
```

Syntax

```
ABE    memgroup [operands]
```

Aliases

```
AB, ABE
```

Defaults

None

Required

None

Operands

memgroup	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
operands	Optional. Can include any TSO ABE operands.

Remarks

ABE was written in PL/I and the PL/I Optimizing Compiler Transient Library is required for its execution.

This interface is optional. Use it only if your installation installed the public domain ABE command.

ACFCOMP Subcommand

The ACFCOMP subcommand compiles and stores a set of CA-ACF2 access or resource rules.

Example

```
ACFCOMP   acfrule* nostore
```

Syntax

```
ACFCOMP   memgroup [FORCE|NOFORCE ]
                   [LIST|NOLIST   ]
                   [MAXRULE(nnn)  ]
                   [STORE|NOSTORE ]
```

Aliases

ACF, ACFC, ACFCO, ACFCOM, ACFCOMP

Defaults

memgroup, FORCE, LIST, MAXRULE(250), STORE

Required

None

Operands

memgroup	Identifies the members to be compiled.
	Default member names, member lists, member name ranges and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
FORCE	Access rule sets are stored, replacing any existing access rule sets with the same keys.
NOFORCE	Access rule sets are not stored or replace any existing access rule sets if they exist with the same keys.
LIST	The rules are listed at the terminal as they are read.
NOLIST	Only compiler control and error messages are listed at the terminal.
MAXRULE(nnn)	Specifies a scaling factor that determines the maximum size of the access rule set that the buffer can hold.
STORE	Access rule sets are stored automatically after compilation.
NOSTORE	Access rule sets are not stored after compilation; this is a test compilation.

Remarks

This interface is optional; use it only if your installation uses CA-ACF2 for security.

ALIAS Subcommand

The ALIAS subcommand adds an alias name for a member.

Example

ALIAS   mainmem   aliasmem

Syntax

```
ALIAS   memname aliasname
```

Aliases

```
AL, ALI, ALIA, ALIAS
```

Defaults

None

Required

memname aliasname

Operands

memname	Identifies the member to which an alias is to be assigned.
aliasname	Specifies the alias name to be assigned. This alias name becomes the default member name.

Remarks

If the data set is not a load library, use the alias name as an alternate name for the identified member.

If the data set is a load library, and if the alias name is an imbedded external name within the load module, that external name is used as the alias entry point; otherwise, the entry point of the named member is used as the alias entry point.

The ATTRIB subcommand can adjust the entry point of an alias or main load member with its ENTRY(name) keyword.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z DEFAULTS FEATURES
--------------- ISPMODE Session Display ---------------------- ROW 128 OF 143
COMMAND ===>                                            SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804       MEM=ZAPHELP ----------------------
>------>alias zap$ zaphelp
PDS010I The alias has been assigned
PDS103I Entry point at 004418  --  ZAPHELP
>----->map

** MAP          ZAPHELP
ZAP       00000000  00003F06
PCL       00003F08  000000E5
CSOUT     00003FF0  00000428
ZAPHELP   00004418  00001B1C
ASMGASM   00005F38  000007E4
OACNOW    00006720  0000032E


PDS103I Entry point at 00004418  --  ZAPHELP
PDS104I Module length  00006A50  --  27K
PDS066I Member is an alias for: ZAP$
```

NOTE A StarTool FDM ALIAS created for a PDSE object containing private CSECTs will not execute, due to invalid object code that results from a limitation of the z/OS Binder when handling private CSECTs. This appears to be a permanent restriction of the z/OS Binder. Message PDS898E is issued to alert you of the existence of private CSECTs in these objects.

CAUTION! The ALIAS subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection".

ATTRIB Subcommand (for source members)

The ATTRIB subcommand lists and modifies the attributes of a member. If the attributes of a group of members is being updated, StarTool FDM identifies the members to be updated and asks whether or not to continue with the update. Add or modify ISPF statistics or SSI information. Since many of the operands apply either to load members or source members only, ATTRIB is described separately with each set of operands.

Example
ATTRIB   *member1*:*member2* ID(*userid*)

Syntax

```
ATTRIB    memgroup [ALIASINFO|NOALIASINFO]
                   [ALIAS|NOALIAS]
                   [LASTREAD]
                   [ADDSTATS]
                   [ID(who)|USERID(who)]
                   [MOD(mm)]
                   [NONE]
                   [SSI(hexdata)|NOSSI]
                   [UNALIAS]
                   [VER(nn)]
                   [CREATED(yyyy/mm/dd)]
                   [MODIFIED(yyyy/mm/dd)]
                   [TIME(hhmm)]
                   [SIZE(nnnnn)|RESIZE]
                   [INIT(nnnnn)]
                   [MODI(nnnnn)]
```

Aliases

A, AT, ATT, ATTR, ATTRI

Defaults

Alias members: *memgroup*, ALIASINFO depending on the CONTROL setting, NOALIAS

Other members: memgroup, NOALIASINFO, NOALIAS.

Required

None

Operands — source members

memgroup	Identifies the source members whose attributes are to be displayed or updated. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
ALIASINFO	Provides names of alias or main members — this operand does not change member attributes.
NOALIASINFO	Does not provide alias information — this operand does not change member attributes.
ALIAS	Updates associated members (aliases, main members, or apparent aliases).
NOALIAS	Does not update associated members.
LASTREAD	For members with ISPF statistics, formats last read date information — does not change member attributes.
ADDSTATS	Adds ISPF statistics for members with no ISPF or SSI information.
ID(*who***)**	Changes the last modifier's user ID (for members with ISPF statistics). Masking to retain current user ID characters may be specified with a **%**.
USERID(*who***)**	Changes the last modifier's user ID (for members with ISPF statistics). Masking to retain current user ID characters can be specified with a **%**.
MOD(mm)	Changes the ISPF modification level (for members with ISPF statistics).
NONE	Removes ISPF statistics or SSI information.
SSI(*hexdata***)**	Adds or changes the SSI information for the member. If you enter less than 8 hexadecimal digits, leading digits on the left are assumed. Members with ISPF statistics cannot have SSI information.
NOSSI	Removes SSI information.
UNALIAS	Converts entry to a main member. **NOTE** If a main entry exists before this conversion, create an "apparent alias". You can detect this situation by a VERIFY subcommand or ATTRIB with the ALIASINFO keyword.
VER(nn)	Changes the ISPF version number (for members with ISPF statistics).
CREATED (yyyy/mm/dd)	Changes the ISPF creation date (for members with ISPF statistics).
MODIFIED (yyyy/mm/dd)	Changes the ISPF modification date (for members with ISPF statistics).
TIME(hhmm)	Changes the ISPF modification time (for members with ISPF statistics).
SIZE(nnnnn)	Changes the ISPF member size (for members with ISPF statistics).
RESIZE	Recalculates the member size (for members with ISPF statistics).

INIT(nnnnn)	Changes the ISPF initial size (for members with ISPF statistics).
MODI(nnnnn)	Changes the ISPF modified size (for members with ISPF statistics).

Remarks — Source Members

If the member is an alias, -A displays after the member name. Convert an alias to a main member with the UNALIAS keyword. If you specify the keyword ALIASINFO, the name of any associated members (aliases, the main member, or any apparent aliases) displays for each member. Otherwise, associated member information displays for *alias* members if CONTROL ALIASINFO is in effect.

> **NOTE** This subcommand modifies the data set if you specify any attribute modifications. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. See Appendix B, "Update Protection".

If the member has ISPF statistics or SSI information, this data is deleted with the NONE keyword.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z DEFAULTS  FEATURES
----------------- ISPMODE Session Display ----------------------  ROW 673 OF 685
COMMAND ===>                                                      CROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=D:DEL  ----------------------
>------>at d:del
PDS230I ISPF Stats: VER.MOD  CREATED    LAST MODIFIED   SIZE  INIT   MOD    ID
PDS230I  DAFJCL      01.00   02/06/15  02/06/15 10:09   25    25     0    C91147
PDS230I  DAFTAL -A   01.00   02/06/15  02/06/15 10:09   25    25     0    C91147
PDS066I Member is an alias for: DAFJCL
PDS230I  DELC        01.02   02/05/10  02/05/10 15:02   19    20     2    C91147
PDS230I  DELINK      SSI: CB296204
PDS230I  DELINK0
PDS117I 2 members counted; cumulative size is 44 records
```

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z DEFAULTS  FEATURES
----------------- ISPMODE Session# 2 Log# 1 --------------------  ROW 847 TO 861 OF 861
COMMAND ===>                                                      SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(A  ---  -------------
>----->attrib = modified(02/04/20)
PDS167I ATTRIB will change the following members: A, ABLK, ADOC, LOADMARK, SAMPSECR, TRIAL601

PDS396A Should ATTRIB continue (Yes/No) ?
>----->y
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
PDS230I  A          01.00   02/04/08  02/04/20 13:21   5     5    0   SER07
PDS230I  ABLK       01.01   02/04/09  02/04/20 14:37   6     9    0   SER07
PDS230I  ADOC
PDS230I  LOADMARK   01.02   02/02/26  02/04/20  6:43  50    48    0   SER07
PDS230I  SAMPSECR   01.26   02/03/27  02/04/20  5:57 197   220  140  SER07
PDS230I  TRIAL601   01.00   02/04/14  02/04/20 15:16 123   123    0  SER07
PDS117I  5 members counted; cumulative size is 381 records

***************************** BOTTOM OF DATA *****************************
```

ATTRIB Subcommand (for load members)

The ATTRIB subcommand lists and modifies the attributes of a member. If the attributes of a group of members is being updated, StarTool FDM identifies the members to be updated and asks whether or not to continue with the update. Since many of the operands apply either to load members or source members only, ATTRIB is described separately with each set of operands.

Example

```
ATTRIB    member1:member2  RENT  REUS  REFR
```

Syntax

```
ATTRIB    memgroup [ALIASINFO|NOALIASINFO]
                   [ALIAS|NOALIAS]
                   [LKEDDATE|NOLKEDDATE]
                   [SHORT]
                   [AMODE24|AMODE31|AMODE64|AMODEANY]
                   [AUTH|NOAUTH]
                   [DC|NODC]
                   [EDIT|NOEDIT]
                   [ENTRY(name)]
                   [EXEC|NOEXEC]
                   [LOADONLY|NOLOADONLY]
                   [NONE]
                   [PAGE|NOPAGE]
                   [REFR|NOREFR]
                   [RENT|NORENT]
                   [REUS|NOREUS]
                   [RLDFIX|NORLDFIX]
                   [RMODE24|RMODEANY]
                   [SSI(hexdata)|NOSSI]
                   [UNALIAS ]
```

Aliases

```
A, AT, ATT, ATTR, ATTRI
```

Defaults

memgroup, ALIAS, LKEDDATE (depending on the CONTROL setting), ALIASINFO (depending on the CONTROL setting)

Required

None

Operands — Load Members

memgroup	Identifies the load members whose attributes are to be displayed or updated. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
ALIASINFO	Provides the names of alias or main members — this operand does not change member attributes.
NOALIASINFO	Does not provide alias information — this operand does not change member attributes.
ALIAS	Updates associated members (aliases, main members, or apparent aliases).
NOALIAS	Does not update associated members.
LKEDDATE	Provides the date of the last linkage edit — does not change member attributes.
NOLKEDDATE	Does not provide the date of the last linkage edit — does not change member attributes.
SHORT	Provides output in an alternate short format — does not change member attributes. This output format is used by default for THEN(attrib) or ELSE(attrib). For an example, see the message PDS232I in the STarTool FDM *Messages Guide*. **NOTE** When the SHORT output format is in use, many standard ATTRIB module checks are not performed. An incorrectly created or modified load module may not be detected.
AMODE24	Changes addressing mode to 24.
AMODE31	Changes addressing mode to 31.
AMODE64	Changes addressing mode to 64.
AMODEANY	Changes addressing mode to ANY.
AUTH	Authorizes the module with APF (equivalent to AC=1 in a linkage edit).
NOAUTH	Removes APF authorization (equivalent to AC=0 in a linkage edit).
DC	The module is downward compatible with the level-E linkage editor.
NODC	The module is not downward compatible with the level-E linkage editor.
EDIT	Allows linkage editing of the module.
NOEDIT	Disallows linkage editing of the module.
ENTRY(*name***)**	Changes the entry point address to the external symbol specified. The symbol entered must be present in the load module.
EXEC	Allows execution of the module.
NOEXEC	Disallows execution of the module.
LOADONLY	Disallows use of the module except by a LOAD MACRO.
NOLOADONLY	Allows module use by any means.

NONE	Changes attributes to the following: AMODE24, NOAUTH, NODC, EDIT, EXEC, NOLOADONLY, NOPAGE, NOREFR, NORENT, NOREUS, NOSSI, and RMODE24. If other parameters are coded with NONE, they override any corresponding parameters.
PAGE	Requires page alignment for the module.
NOPAGE	Allows any alignment for the module.
REFR	Adds the refreshable attribute.
NOREFR	Removes the refreshable attribute.
RENT	Adds the reentrant (and reusable) attribute.
NORENT	Removes the reentrant attribute.
REUS	Adds the reusable attribute.
NOREUS	Removes the reusable attribute.
RLDFIX	Changes the directory's RLD/CONTROL count to match the value in the first RLD entry (for modules linked since OS/VS).
NORLDFIX	Does not modify the directory's RLD/CONTROL count.
RMODE24	Changes residence mode to 24 (below the 16-Megabyte line).
RMODEANY	Changes residence mode to ANY (above the 16-Megabyte line).
SSI(*hexdata***)**	Adds or changes the SSI information for the member. If you enter less than 8 hexadecimal digits, leading digits on the left are assumed.
NOSSI	Removes any SSI information for the member.
UNALIAS	Converts entry to a main member. **NOTE** If a main entry exists before this conversion, create an "apparent alias". You can detect this situation by a VERIFY subcommand or ATTRIB with the ALIASINFO keyword.

Remarks — Load Members

The following linkage editor attributes are identified if they are present for a member:

AMODE	Addressing mode (if AMODE is not 24 or RMODE is not 24).
AUTH	APF authorized (or AC=1).
DC	Downward compatible (the module can be reprocessed by a Level-E linkage editor).
LOAD ONLY	Only Loadable (OL — the module can be brought into main storage only with a LOAD MACRO).
NOT EDIT	Not Editable (the module cannot be linkage edited).
NOT EXEC	Not Executable (the linkage editor noted an error; LET processing was not requested).
OVERLAY	Overlays load module structure.
PAGE	Page alignment required.
REFR	Refreshable (replaceable by a new copy during execution).

RENT	Reentrant (executable by several tasks simultaneously).
REUS	Reusable (executable by several tasks in serial order).
RMODE	Residence mode (if RMODE is not 24 or AMODE is not 24).
SCTR	Scatters load module structure.
SSI	SSI information in hexadecimal.
TEST	Linked with the TEST option.

When the ATTRIB subcommand adds or deletes linkage attributes for a member it performs some additional processing based on the member's attributes and the keywords specified.

1 If a changed member is reentrant, the reusable attribute is added (as does the linkage editor).

2 If a main member is assigned an RMODE or AMODE value, any associated alias directory entries are updated to reflect the same RMODE value and the AMODE value of the main member.

3 Alias information is provided unless NOALIASINFO is specified or CONTROL NOALIASINFO is in effect.

   a If the member is an alias, the name of any corresponding main member displays if it exists; otherwise, the member name from the alias directory entry (the main member name at the time the alias was created) displays.

   b If the member is not an alias, the name of any corresponding alias members and any apparent aliases displays.

4 If there is a mismatch between the directory RLD/CONTROL field and the first RLD entry of OS/VS linked modules, StarTool FDM displays an appropriate error message.

> **NOTE** This subcommand modifies the data set if you specify attribute modifications. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by StarTool FDM for SHR allocations. For details, see Appendix B, "Update Protection".

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------ ISPMODE Session Display --------------------  ROW 691 OF 733
COMMAND ===>                                                    SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=PDS*  --------------------------
>----->at pds*
PDS020I PDSPGM Attributes are: RENT, REUS
PDS120I Residence Mode is ANY -- addressing mode is 31
PDS103I Entry point at 00000000  --  WHAT
PDS104I Module length  0006CFA8 --   436K
PDS064I Last link-edited on 02/06/10  by LKED 566528408 V71 M00

PDS020I PDSRX Attributes are: RENT, REUS
PDS103I Entry point at 00000000  --  PDSRX
PDS104I Module length  00003D28 --    16K
PDS064I Last link-edited on 02/05/10  by LKED 566528408 V71 M00

PDS118I 1 members RMODE24; size is 16K
PDS119I 1 members RMODEANY; size is 436K
```

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------- ISPMODE Session# 2 Log# 1 ---------------  ROW 868 TO 889 OF 889
COMMAND ===>                                                SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007          MEM=(#OPCODE -------------------
>----->at = rent
PDS167I ATTRIB will change the following members: #OPCODE, #OPCODEM, #OPCODE2
PDS396A Should ATTRIB continue (Yes/No) ?
>----->y
PDS020I #OPCODE Attributes are: RENT, REUS, REFR
PDS103I Entry point at 00000000  --  OPTESTCD
PDS104I Module length  000008E0  --      3K
PDS064I Last link-edited on 02/12/20  by LKED 566528408  V03 M01

PDS020I #OPCODEM Attributes are: RENT, REUS, REFR, NOT EXEC
PDS103I Entry point at 00000000  --  OPTESTCD
PDS104I Module length  00000E00  --      4K
PDS064I Last link-edited on 02/12/20  by LKED 566528408  V03 M01

PDS020I #OPCODE2 Attributes are: RENT, REUS, REFR
PDS103I Entry point at 00000000  --  OPTESTCD
PDS104I Module length  000003B8  --      1K
PDS064I Last link-edited on 02/12/20  by LKED 566528408  V03 M01

PDS118I 3 members RMODE24; size is 7K

***************************** BOTTOM OF DATA *********************************
```

In the screen display above, the LKED value in message PDS046I is replaced with a coded value to represent the actual linkage editor or program binder used for the listed CSECT. StarTool FDM determines this through inspection of the IBM component codes carried in the load module. For more information about the IBM component codes, see message PDS046I in the StarTool FDM *Messages Guide*.

BLK3380 Subcommand

The BLK3380 subcommand computes an optimal block size for placing a data set on a 3380 disk pack. The program output includes the following reports:

1 A summary block size report for the given LRECL and key length that includes the recommended block size to use.

2 A recommended data set space allocation.

3 An optional track capacity report for the provided key length.

Example

BLK3380 133

Syntax

```
BLK3380   lrecl    [KEYLENGTH(keylen)           ]
                   [TRACKCAP | NOTRACKCAP        ]
                   [BLKSIZE(blklen)             ]
```

```
[NUMBER(#rec) | RECORDS(#rec)  ]
[VERIFY                        ]
```

Aliases

BLK338, BLK3380

Defaults

KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required

lrecl

Operands

lrecl	The logical record length of the data that is to be placed in the data set.
KEYLENGTH(kl)	The key length, in bytes, of the keys to be used in the data set. The maximum legal key length is 255.
TRACKCAP	Specifies a track capacity report for the device using the specified (or default) key length. A track capacity report is provided if NOTRACKCAP is not specified and lrecl exceeds the maximum block size for a track or BLKSIZE exceeds the maximum block size for a track.
NOTRACKCAP	Specifies that a track capacity report is not wanted.
BLKSIZE(blklen)	The block size to use for the allocation computation. If you do not enter blklen (or you enter zero), the program's recommended block size is used.
NUMBER(#rec)	Number of logical records in the data set.
RECORDS(#rec)	Number of logical records in the data set.
VERIFY	Specifies that the TRKCALC routine is to be used to verify track capacity calculations. If you use VERIFY, the number of calls to TRKCALC is output at the end of the output. With VERIFY on, you need a minimum of 34 calls for a track capacity table; otherwise, you need a minimum of 17 calls for the track capacity table.

Remarks

The recommended block size value is for data sets in which the predominant access is sequential. For data sets where random access time is critical or the usual access is random, use a small block size (500 - 2000 bytes).

The recommended block size tends to be near a half-track figure since this is the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs, and overall processing time. This figure is only a

general guide; for maximum efficiency considering other factors, study the generated block size summary report or a track capacity report.

The program assumes a large amount of data is to be stored. Place data sets that occupy a few tracks in partitioned data sets. If not possible, use a small block size (2400 - 4000 bytes).

This interface is optional. Use it if your installation installed the public domain BLK3380 subcommand.

```
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY --------------------------
>----->blk3380 80 tr
  3380 BLOCKSIZE SUMMARY;    LRECL=80        KEY LENGTH=0
          BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
          ---------    ------------    ------------    -----------
                 80              83              83          14.0%
              2,480              16             496          83.6%
              . . . .
              7,440               6             558          94.0%
              9,040               5             565          95.2%
             11,440               4             572          96.4%
             15,440               3             579          97.6%
 RECOMMENDED-->23,440             2             586          98.7%
             32,720               1             409          68.9%

FOR BLKSIZE 23,440  AND   100,000 RECORDS, ALLOCATE:
      342 BLOCKS,        171 TRACKS, OR        12 CYLINDERS

  3380 TRACK CAPACITY;    KEY LENGTH=0
      BLOCKS/TRACK      BLKSIZE      BYTES/TRACK      UTILIZATION
      ------------      -------      -----------      -----------
                 1       47,476           47,476         100.0%
                 2       23,476           46,952          98.9%
                 3       15,476           46,428          97.8%
                 4       11,476           45,904          96.7%
                 5        9,076           45,380          95.6%
                 6        7,476           44,856          94.5%
              . . . .
                16        2,484           39,744          83.7%

DEVICE SUMMARY: MAX BLOCKSIZE=47,476    TRACKS=13,275    BYTES=630,243,900
******************************** BOTTOM OF DATA********************************
```

BLK3390 Subcommand

The BLK3390 subcommand computes an optimal block size for placing a data set on a 3390 disk pack. The program output includes the following reports:

1   A summary block size report for the given LRECL and key length that includes the recommended block size to use.

2   A recommended data set space allocation.

3   An optional track capacity report for the provided key length.

Example

```
BLK3390 121
```

Syntax

```
BLK3390   lrecl   [KEYLENGTH(keylen)               ]
                  [TRACKCAP | NOTRACKCAP            ]
                  [BLKSIZE(blklen)                 ]
                  [NUMBER(#rec) | RECORDS(#rec)    ]
                  [VERIFY                          ]
```

Aliases

```
BLK339, BLK3390
```

Defaults

KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required

lrecl

Operands

lrecl	The logical record length of the data to be placed in the data set.
KEYLENGTH(kl)	The key length, in bytes, of the keys to be used in the data set. The maximum legal key length is 255.
TRACKCAP	A track capacity report for the device using the specified (or default) key length. A track capacity report is provided if NOTRACKCAP is not specified and lrecl exceeds the maximum block size for a track or BLKSIZE exceeds the maximum block size for a track.
NOTRACKCAP	Do not provide a track capacity report.
BLKSIZE(blklen)	The block size to use for the allocation computation. If you do not enter blklen (or you enter zero), the program's recommended block size is used.
NUMBER(#rec)	Number of logical records in the data set.
RECORDS(#rec)	Number of logical records in the data set.
VERIFY	Use the TRKCALC routine to verify track capacity calculations. If you use VERIFY, the number of calls to TRKCALC is output at the end of the output. With VERIFY on, you need a minimum of 34 calls for a track capacity table; otherwise, you need a minimum of 17 calls for the track capacity table.

Remarks

The recommended block size value is for data sets in which the predominant access is sequential. For data sets where random access time is critical or the access is random, use a small block size (500 - 2000 bytes).

The recommended block size tends to be near a half-track since this is considered the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs, and overall processing time. This is only a general guide; for maximum efficiency, considering other factors, study the generated block size summary report or a track capacity report.

The program assumes a large amount of data to be stored. Place data sets that occupy only a few tracks in partitioned data sets. If this is not possible, use a small block size (2400 - 4000 bytes).

This interface is optional; use it if your installation installed the public domain BLK3390 subcommand.

```
COMMAND ===>                                               SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  --------------------------
>----->blk3390 80 tr
  3390 BLOCKSIZE SUMMARY;    LRECL=80         KEY LENGTH=0
            BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
            ---------    ------------    ------------    -----------
                   80              78              78         11.0%
                2,880              16             576         81.3%
                 . . . .
                8,880               6             666         94.0%
               10,720               5             670         94.6%
               13,680               4             684         96.6%
               18,400               3             690         97.4%
 RECOMMENDED-->27,920               2             698         98.5%
               32,720               1             409         57.7%

FOR BLKSIZE 27,920  AND   100,000 RECORDS, ALLOCATE:
      287 BLOCKS,        144 TRACKS, OR        10 CYLINDERS

  3390 TRACK CAPACITY;    KEY LENGTH=0
       BLOCKS/TRACK     BLKSIZE      BYTES/TRACK     UTILIZATION
       ------------     -------      -----------     -----------
                  1      56,664           56,664         100.0%
                  2      27,998           55,996          98.8%
                  3      18,452           55,356          97.7%
                  4      13,682           54,728          96.6%
                  5      10,796           53,980          95.3%
                  6       8,906           53,436          94.3%
                . . . .
                 16       2,942           47,072          83.1%

DEVICE SUMMARY: MAX BLOCKSIZE=56,664   TRACKS=16,695   BYTES=946,005,480
NOCYLS=1,113   TRKS/CYL=15   TRKSIZE=58,786   DSCB/TRK=50   PDS/TRK=45
***************************** BOTTOM OF DATA *******************************
```

BLK9345 Subcommand

The BLK9345 subcommand computes an optimal block size for placing a data set on a 9345 disk pack. The program output includes the following reports:

1   A summary block size report for the given LRECL and key length that includes the recommended block size to use.

2   A recommended data set space allocation.

3   A optional track capacity report for the provided key length.

Example
BLK9345 80

Syntax
BLK9345   lrecl   [KEYLENGTH(keylen)   ]

```
[TRACKCAP | NOTRACKCAP        ]
[BLKSIZE(blklen)              ]
[NUMBER(#rec) | RECORDS(#rec) ]
[VERIFY                       ]
```

Aliases

BLK9, BLK93, BLK934, BLK9345

Defaults

KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required

lrecl

Operands

lrecl	The logical record length of the data that is placed in the data set.
KEYLENGTH(kl)	The key length, in bytes, of the keys used in the data set. The maximum legal key length is 255.
TRACKCAP	A track capacity report for the device using the specified (or default) key length. A track capacity report is provided if NOTRACKCAP is not specified and lrecl exceeds the maximum block size for a track or BLKSIZE exceeds the maximum block size for a track.
NOTRACKCAP	Do not provide a track capacity report.
BLKSIZE(blklen)	The block size to use for the allocation computation. If you do not enter blklen (or if you enter zero), the program's recommended block size is used.
NUMBER(#rec)	Number of logical records in the data set.
RECORDS(#rec)	Number of logical records in the data set.
VERIFY	Use the TRKCALC routine to verify track capacity calculations. If you use VERIFY, the number of calls to TRKCALC for a track capacity table is output at the end of the output. With VERIFY on, you need a minimum of 34 calls for a track capacity table; otherwise, you need a minimum of 17 calls for the track capacity table.

Remarks

The recommended block size value is for data sets in which the predominant access is sequential. For data sets where random access time is critical or the access is random, use a small block size (500 - 2000 bytes).

The recommended block size tends to be near a half-track figure since this is considered the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs, and overall processing time. This is only a general guide; for maximum efficiency considering other factors, study the generated block size summary report or a track capacity report.

The program assumes that a large amount of data is to be stored. Place data sets that occupy only a few tracks in partitioned data sets. If this is not possible, use a small block size (2400 - 4000 bytes).

This interface is optional; use it if your installation installed the public domain BLK9345 subcommand.

```
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY   ------------------------
>----->blk9345 80 tr
  9345 BLOCKSIZE SUMMARY;    LRECL=80        KEY LENGTH=0
          BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
          ---------    ------------    ------------    --------
              2,240           16             448           77.1%
                . . . .
              7,200            6             540           93.0%
              8,800            5             550           94.7%
             11,120            4             556           95.7%
             15,040            3             564           97.1%
 RECOMMENDED-->22,880            2             572           98.5%
             32,720            1             409           70.4%

FOR BLKSIZE 22,880  AND   100,000 RECORDS, ALLOCATE:
      350 BLOCKS,        175 TRACKS, OR        12 CYLINDERS

  9345 TRACK CAPACITY;    KEY LENGTH=0
        BLOCKS/TRACK     BLKSIZE     BYTES/TRACK     UTILIZATION
        ------------     -------     -----------     -----------
                   1      46,456         46,456         100.0%
                   2      22,928         45,856          98.7%
                   3      15,074         45,222          97.3%
                   4      11,158         44,632          96.1%
                   5       8,810         44,050          94.8%
                   6       7,214         43,284          93.2%
              . . . .
                  16       2,314         37,024          79.7%

DEVICE SUMMARY: MAX BLOCKSIZE=46,456   TRACKS=21,600   BYTES=1,003,449,600
NOCYLS=1,440    TRKS/CYL=15    TRKSIZE=48,280   DSCB/TRK=45    PDS/TRK=40
**************************** BOTTOM OF DATA ********************************
```

BROWSE Subcommand

The BROWSE subcommand enters ISPF browse for a member.

Example

```
BROWSE    mema:memb
```

Syntax

```
BROWSE   memgroup
      * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)   ]
      * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)          ]

 *NOTE: Lines with an asterisk are supported for VSAM with PBROWSE or
     BRIF.
```

Aliases

B, BR, BRO, BROW, BROWS, BROWSE

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER

Defaults

memgroup

Required

None

Operands

memgroup	Identifies the members to be browsed. Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
FROMKEY(ky)	For VSAM data sets only, ky is the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. It can be specified only for an alternate index or a key-sequenced data set.
FROMADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Code only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. It can be specified only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is the key of the last record to be accessed. This is a generic key and can be coded as 'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. It can be specified only for an alternate index or a key-sequenced data set.
TOADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Code only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if the spanned records are to be accessed

TONUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.
	Use this parameter with FROMNUMBER. It can be specified only for a variable or fixed relative record data set.

Remarks

If you enter a **:** for the member name position, an ISPF member selection list displays.

Display VSAM data sets with PBROWSE. You may have chosen a different interface during StarTool FDM installation. Of the other interfaces, only BRIF supports the positioning keywords such as FROMKEY. To determine the interface, type a CONTROL DEFAULT subcommand and look for an output line beginning BROWSE calls with one of the following values:

- BRIF – uses the ISPF BRIF interface which is up to 32760 characters wide

- PBROWSE – uses PBROWSE services

- REVIEW – uses the REVIEW command (a non-ISPF full screen public domain program)

- %VSAMMBR – uses the MacKinney System VSAM Utility Browse

If the ISPF BRIF interface is in use, StarTool FDM keeps the records of a data set in high region storage if the data set size does not exceed two megabytes or the data set is a path (since a path cannot have unique keys). Records are provided to the BRIF service as requested. If you "max down" to the end of the data set, this could take a considerable amount of time. If the data set is in storage, subsequent positioning in the browse is much faster.

CALC Function

The CALC command performs floating point calculations.

Example
CALC

Syntax
CALC

Aliases
CAL, CALC

Operands

No operands are supported for the CALC subcommand.

Remarks

The CALC ISPF table displays in response to a CALC command. When you are in a CALC display, you have many options. Delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, and so on. For assistance with the different options available, use the HELP command, CUA functions, or type an O command as a primary command.

The CALC table displays the calculator tape for your calculator. Enter new calculations from the CALC display.

The following primary commands are supported directly for the CALC function. For information on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

EDITT[BL] (or ET[BL])	Enters an edit session on CALC table data.	
F	Finds a string and positions the display start location.	
	Syntax	`F anystring [ASIS]` `    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
OUT[PUT]	Outputs the CALC table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the CALC table based on a string match.	
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD][NOT]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
X	Clears the CALC table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the CALC table; this is equivalent to X ALL	

CAX Function

The CAX command displays the names and attributes of active system catalogs in an ISPF table.

Example
CAX

Syntax
CAX

Aliases
CAX, CAXW, CAXWA

Operand

No operands are supported for the CAX command.

Remarks

The CAX ISPF table displays in response to a CAX command. When you are in a CAX display, you have many options. Delete a part of the table, sort the table in different directions, find data in the table, print a part of the table, store a part of the table in a data set, and so on. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as a primary command or as a line command.

The CAX table displays the names of all active catalogs for your system and allows you to enter LISTCAT-type line commands against them.

The catalogs are open at the time of the list. Use the information to fix suspected broken catalogs. StarTool FDM does not fix catalogs, but the display of information on the open catalogs aid in the diagnosis of the problems.

The results of line commands against the CAX list are captured in the StarTool FDM log. Save them for later sessions in an alternate log.

The CAX table displays the active system catalogs and information about each catalog on a line level: Type, Status, CAXWA address, relative entry number, volume name, and data set name.

Type	Catalog type:
	■ ICF - Integrated Catalog Facility type catalog
	■ VSAM - old type VSAM catalog
	■ RECV - old type VSAM recoverable catalog
	■ MSTR - master catalog
Status	Catalog status:
	■ ACTIVE - usual catalog status
	■ DELETE - a catalog delete occurred but the catalog is still allocated; it will be deleted when freed by all users
	■ DISCON - a catalog DISCONNECT occurred but the catalog is still allocated; it will be disconnected when freed by all of its users
	■ HUNG - a hung flag was set for this catalog
	■ -IOE - a permanent I/O error was detected for this catalog
	■ -NOM - no memory was available for catalog processing
Address	Address of the CAXWA control block.
Entry	Relative entry number of each CAX entry.
Volume	Volume on which the catalog resides.
Dsname	Data set name of the catalog.
	The following primary commands are supported directly for the CAX function; for documentation on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
CAX	Rebuilds the CAX table after you empty it with the XALL command).	
EDITT[BL] (or ET[BL])	Enters an edit session on CAX table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax	`F anystring [ASIS]`
		`    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
		`    [ANY|DSNAME|VOLUME]`

L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax	`LOCATE   number|dsname|volume`
O[PTIONS]	Provides primary command selection for CAX and operand syntax assistance.	
OUT[PUT]	Outputs the CAX table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the CAX table based on a string match.	
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]` `        [ANY|DSNAME|VOLUME]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are set to RFIND.	
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [NUMBER|DSNAME|VOLUME]` `     [ASCEND|DESCEND]`
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as by your installation.	
X	Clears the CAX table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the CAX table; this is equivalent to **X ALL**	

The CAX function supports following line commands:

+	Provides an extension panel for command entry. *cm ERR- t="Line command,=">
=	Repeats the previous line command. *cm ERR- t="Line command,=">

IDC	Issues an IDCAMS LISTC and directs the output to the log.
K	Kills and clears all following line commands.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
O	Provides line command selection and operand syntax assistance.
S	Selects line command (normally UT, it is set by SETSEL).
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
UT	Selects the extended user line command panel.
X	Drops the table line.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The CAX function supports the following block line commands:

==	= command. Repeats the previous line command for the range of lines. *cm ERR- t="Block line command,==">
LL	LOG command. Copies the range of lines into the log.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
XX	X command. Drops the range of table lines.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------ Active Catalog Display ----------------------  ROW 1 OF 20
COMMAND ===>                                               SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=:  --------------------------------
CMD  DATA/MSG TYPE STATUS -ADDR- NUM VOLUME ---------DATA SET NAME ------
            ICF  ACTIVE A5B058   1 SPD801 CATALOG.ISPD801.CNA730
            ICF  ACTIVE A6E118   2 TESS05 CATALOG.ITESS05.CN3307
            ICF  ACTIVE A80100   3 STR502 CATALOG.ISTR502.CNA800
            ICF  ACTIVE A8F030   4 IBNK80 CATALOG.IIBNK80
            ICF  ACTIVE A95058   5 TES806 CATALOG.ITES806.CN3196
            ICF  ACTIVE AC1990   6 DDA500 CATALOG.IDDA500.CN3440
            ICF  ACTIVE AC8DB8   7 AION81 CATALOG.IAION81.CN3871
            ICF  ACTIVE AC8628   8 TES802 CATALOG.ITES802.TEMP
            ICF  ACTIVE AC86E0   9 ACCT01 CATALOG.IACCT01.CNA900
            ICF  ACTIVE AF6AD8  10 GLD800 CATALOG.IGLD800.CN3881
            ICF  ACTIVE AC8968  11 STR516 CATALOG.ISTR516.CNA800
            ICF  ACTIVE AC1030  12 FICO80 CATALOG.IFICO80.CN9100
            ICF  ACTIVE AD8280  13 STR501 CATALOG.ISTR501.CNA800
            ICF  ACTIVE AC83C8  14 STR504 CATALOG.ISTR504.CNA800
            ICF  ACTIVE AC8218  15 IAC800 CATALOG.IIAC800.CN2650
            ICF  ACTIVE AC8A80  16 STR503 CATALOG.ISTR503.CNA800
            ICF  ACTIVE AF68E0  17 STR505 CATALOG.ISTR505.CNA800
            ICF  ACTIVE AF7118  18 STR803 CATALOG.ISTR803.CNA850
```

CHANGE Subcommand

The CHANGE subcommand switches StarTool FDM to a different data set.

Example

```
CHANGE   lib.cntl
```

Syntax

```
CHANGE   {dsn [VOLUME(vser)|VOLSET(vdef)] [SHR|OLD] |
          * | FILE(ddname) [NUMBER(num)]}
```

Aliases

STAR, START, STARTO, STARTOO, STARTOOL, C, CH, CHA, CHAN, CHANG, CHANGE

Defaults

SHR or previously used data set name and volume if no operand or ***** is entered.

Required

None

Operands

dsn	Identifies the data set name. If the data set name is not entered in single quotes ('), your TSO PREFIX is appended to the start of the entered data set name. If you enter ***** in the data set name position, the previous set name is assumed (you are switched from the current data set to the previous data set with the ***** operand; another CHANGE * switches data sets back again). If your system has password security, type your data set password after the data set name followed by a slash (/). The syntax is dsn/password.
VOLUME(vser)	Specifies the volume name on which the data set resides. If you type a volume name, SYSALLDA is assumed as the unit name; otherwise, use the unit name from the catalog. Use this parameter if the data set is not cataloged or if the catalog entry is not to be used. To use the catalog for a CHANGE subcommand even if VOLSET is in effect, type VOLUME(*).
VOLSET(vdef)	Specifies a default volume name for data set references. For example, if you type VOLSET(NEWRES), subsequent CHANGE subcommands with a data set name and no VOLUME parameter assume a VOLUME(NEWRES) parameter. Use this parameter if you want a default volume. To nullify the effect of VOLSET, type VOLSET(*) on a subsequent CHANGE subcommand.
SHR	Allocates the data set with a disposition of SHR; allows simultaneous use of this data set by others. SHR is recommended.
OLD	Allocates the data set with a disposition of OLD; does not allow simultaneous use of this data set by others. SHR is recommended.
FILE(ddname)	Identifies the DDNAME of a pre-allocated data set. Only disk data sets (including VIO) are supported. If you use the FILE keyword, do not use dsn, SHR/OLD, VOLSET, and VOLUME. However, if the data set is concatenated, use the SHR or OLD keyword since StarTool FDM reallocates the data set.
NUMBER(num)	Specifies the concatenation number for the DDNAME allocation for the FILE keyword. *num* defaults to 1 but if *num* is larger than the number of concatenated data sets, the last data set in the sequence is used.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ---------- ROW 430 OF 434
COMMAND ===>                                          SCROLL ===> CSR
- DSN=WESTERN.RGN.LINK,VOL=SER=STR815  MEM=:  --------------
>----->c 'western.rgn.link'
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80   9040     1X    47      10    40 TRK        2

PDS223I This is a linklist data set; all linklist libraries are authorized
PDS189I This data set is managed by LLA
PDS224I This data set is APF authorized
PDS226I This data set has 2 free directory blocks
PDS298I There are 4 users allocated to this data set

PDS300A ENTER OPTION -- DSN=WESTERN.RGN.LINK,VOL=SER=STR815  MEM=:

******************************** BOTTOM OF DATA  ***************************
```

CMDTBL Function

The CMDTBL function manages ISPF command tables. CMDTBL displays command tables (by default, the current ISP command table is selected from the copy in memory) and manipulates command table entries. Move, modify, and test Individual command table entries in the CMDTBL function before saving the table.

The CMDTBL function builds, merges, or changes the edit mode of the command table. After the CMDTBL function begins, shift left or right to see a double-line view of the command table entries.

A command table contains the specifications of general commands that you can enter from any panel during the execution of an application. Command table entries are identified by an application ID. They are maintained in the ISPF table input library (ISPTLIB).

Example
```
CMDTBL    isp edit
```

Syntax
```
CMDTBL    [appl | ISP | ISR | APPLTBL | SITETBL | USERTBL ]
          [EDIT | NOEDIT                 ]
          [NEW | TOP | BOTTOM | INSERT ]
          [LIB(ddname)                 ]
```

Aliases
```
CMD, CMDT, CMDTB, CMDTBL
```

Defaults

ISP, NOEDIT, INSERT, LIB(ISPTLIB)

Required

None

Operands

appl	A 1- to 4-character application name. ISP, ISR, APPLTBL, SITETBL, and USERTBL applications are handled as special cases. The associated table is retrieved from memory if it is active.
ISP	Invokes CMDTBL with the ISP application.
ISR	Invokes CMDTBL with the ISR application.
APPLTBL	Invokes CMDTBL with the active application (normally, ISP or ISR).
SITETBL	For ISPF 4.2 or above, invokes CMDTBL with the site-defined application.
USERTBL	For ISPF 4.2 or above, invokes CMDTBL with the user-defined application.
EDIT	Edits the command table.
NOEDIT	Displays the command table.
NEW	Creates a new command table.
TOP	Adds a command table to the top of the current command table.
BOTTOM	Adds a command table after the end of the current command table.
INSERT	Inserts a command table after the top line being displayed of the current command table.
LIB(ddname)	*ddname* to use for reading the table library (this defaults to ISPTLIB).

Remarks

If CMDTBL is already active, the default APPL name is the current name and no merge is done. Use with EDIT or NOEDIT to change the mode. For ISP, ISR, APPLTBL, SITETBL, or USERTBL, LIB is ignored and the in-memory copy is used.

Type an ISPF command using any of the following methods:

■   In a COMMAND field followed by Enter

■   Press a PF key

■   Select an attention field with a light pen or cursor select key

After entering a command, ISPF searches the application command table (if one exists) and then the system command table (ISP or ISR). If it finds the command, ISPF takes the proper action immediately; otherwise, ISPF passes the command through to the dialog without changes in the COMMAND field.

The CMDTBL function supports the following table entries:

CMD	Specifies line commands that affect this table entry.
DATA/MSG	Provides feedback on commands.
VERB	Specifies the name of the command table entry (can contain 2 to 8 characters).
T (for Truncation)	Specifies the minimum number of characters that you must enter to find a match with the command verb (0 means that no truncation is allowed and 1 is not supported).
ACTION	Specifies the ISPF action to be performed when the command is entered (can contain up to 60 characters). The following actions are supported:
	■ SELECT - followed by SELECT keywords, causes a dialog (command, program, or panel) to be given control. Check this type of entry with the TEST line command. The &ZPARM parameter is often used in a SELECT entry. &ZPARM is replaced by any parameters entered with the command.
	■ ALIAS - followed by the name of another command and an optional parameters, specifies a command alias.
	■ PASSTHRU - causes the command to be passed through to the dialog (as if it were not in the table).
	■ SETVERB - causes the command to be passed through to the dialog with the command verb stored separately from the parameters.
	■ NOP - causes the command to be inactivated. An inactive command message displays for this type of command.
	■ blank - causes the command table entry to be ignored and continues scanning to search for additional entries with the same command verb.
	■ &name - allows dynamic specification of a command table entry. Specify any ISPF variable, but a leading & is required.
DESCRIPTION	Contains a brief description of a command (can contain up to 57 characters).

The following primary commands are supported directly for the CMDTBL function; for documentation on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
CL[OSE]	Closes the command table application with no changes.	
EDITT[BL] (or ET[BL])	Enters an edit session on the command table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax	`F anystring [ASIS]`
		`    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
		`    [ANY|VERB|ACTION|DESCRIPT]`

IN[SERT]	Adds 10 blank lines to the current command table after the current row displayed at the top of the screen.
O[PTIONS]	Provides primary command selection for the CMDTBL function and operand syntax assistance.
OUT[PUT]	Outputs the command table to print or a data set.
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the command table based on a string match.
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]` `        [ANY|VERB|ACTION|DESCRIPT]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.
SA[VE]	Updates the table to a data set or in memory.
	Syntax	`SAVE appl	ISP	ISR` ` [REPLACE] [NOCHECK] [LIB(ddname)]`
	Defaults: current appl, LIB(ISPTABL) For ISP or ISR and REPLACE, the in-memory copy is always updated; NOCHECK avoids command table validity checking.	
NOS[SAVE]	Closes the command table without making any command table updates.
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
	Syntax	`TAG linecmd`
X	Clears the command table relative to the cursor position.
	Syntax	`X [ABOVE	BELOW	ALL]`
XA[LL]	Clears the command table; this is equivalent to **X ALL**

The following line commands are supported in the CMDTBL function:

=	Repeats the previous line command. *cm ERR- t="Line command,=">
A	After this entry.
B	Before this entry.
C	Copies line.
DEL	Deletes line.
In	Inserts line with an optional count.
K	Kills and clears any following line commands.
LOG	Copies the line into the log.
M	Moves line.
O	Provides line command selection assistance.
Rn	Repros line with an optional count.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
TEST	Tests the command table entry with prompting if required.

Xn	Drops the command table entry with an optional count.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

==	= command. Repeats the previous line command for the range of lines. *cm ERR- t="Block line command,==">
CC	C command. Copies block.
DD	D command. Deletes block.
MM	M command. Moves block.
RR	R command. Repros block.
XX	X command. Drops the range of table lines.

```
------------- ISPF Command table display of ISP ------------ ROW 10 TO 28 OF 57
COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command or PDSTOOL command
   ----------------------------------------------------------------------------
CMD  DATA/MSG Verb     Description
              LISTC    PDS/USER CONTROLLED PASSTHRU
              PLIST    StarTool PLIST facility
              TOP      SCROLL TO TOP
              BACKWARD SCROLL UP
              UP       SCROLL UP
              BOTTOM   SCROLL TO BOTTOM
              TRYPAN   SELECT A PANEL
              DOWN     SCROLL DOWN
              LEFT     SCROLL LEFT
              RIGHT    SCROLL RIGHT
              RETURN   RETURN
```

```
--------------- ISPF Command table display of ISP ------------OW 10 TO 17 OF 57
COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command or STARTOOL command,  Enter SAVE to save the table

   ----------------------------------------------------------------------------
CMD VERB      T  ACTION
     DATA/MSG     DESCRIPTION

    LISTC     0  &PDSPASS
----              PDS/USER CONTROLLED PASSTHRU
    PLIST     2  SELECT CMD(STARTOOL F(ISPPROF) ISPXEQS LISTC 20 PROMPT
----              StarTool PLIST facility
    TOP       0  ALIAS UP MAX
----              SCROLL TO TOP
    BACKWARD  0  ALIAS UP
----              SCROLL UP
    UP        0  SETVERB
----              SCROLL UP
    BOTTOM    0  ALIAS DOWN MAX
----              SCROLL TO BOTTOM
    TRYPAN    0  SELECT PANEL(&ZPARM)
----              SELECT A PANEL
```

COMBINE Subcommand

The COMBINE subcommand joins members together in an IEBUPDTE format. Any ISPF statistics are preserved and a ./ in the input members is translated to >< in the output. COMBINE supports record lengths beyond 256 characters.

Example

```
COMBINE   pep*  output.data(savmemb)
```

Syntax

```
COMBINE   memgroup
          dsname [VOLUME(volser)]
          [IEBUPDTE|TITLE|NOSEP      ]
          [SEPCHAR(dd)               ]
          [TRANSFROM(ab)             ]
          [TRANSTO(cd)               ]
          [CALL|NOCALL               ]
          [SUMMARY|LIST|NOLIST       ]
          [OLD|SHR | MOD | NEW [BLK|TRK|CYL]
                              [DIR(num)]   [SPACE sec)]]
                                [ROUND]   [EXPDT(yyddd)|RETPD(nnnn)]
      [RLSE]
                              [STORCLAS(sclass)]   [MGMTCLAS(mclass)]
                              [DATACLAS(dclass)]   [DSNTYPE(LIBRARY|PDS)]
```

Aliases

COMB, COMBI, COMBIN, COMBINE

Defaults

memgroup, SHR, IEBUPDTE, SEPCHAR(./)

If NEW is specified, BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE default to values used for the input data set.

Required

memgroup, dsname

Operands

memgroup	Identifies the group of members to be joined together. Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
dsname	Specifies the output data set (and member if partitioned) using standard TSO syntax.
VOLUME(volser)	For NEW data sets, specifies the output volume name; for existing data sets, use this parameter if the output data set is not cataloged. For new data sets, code VOLUME(*) to mean the volume parameter is to be copied.
IEBUPDTE	Records output in IEBUPDTE format (add a "./ ADD NAME=" statement before each output member and preserve ISPF statistics).
TITLE	Places a title line between output members giving the member and data set name.
NOSEP	Places no separators between output members.
SEPCHAR(dd)	Names the IEBUPDTE type separator to place between members (usually ./).
TRANSFROM(ab)	Defaults to ./. A character string is translated to the TRANSTO string if found in column one of the input stream.
TRANSTO(ab)	Defaults to <>. A character string is substituted for the TRANSFROM string if found in column one of the input stream.
CALL	Invokes the copy program.
NOCALL	Do not invoke the copy program; member checking and data set allocations are still performed.
SUMMARY	Display only summary and error messages.
LIST	Display all messages.
NOLIST	Display no messages.
SHR	Allows shared use of the output data set; StarTool FDM provides output data set protection. See Appendix E Appendix B, "Update Protection".
OLD	Obtains exclusive use of the output data set.
MOD	For sequential output, obtains exclusive use and adds to the end of the data set.
NEW	Creates the output data set. When you specify NEW, specify BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT (or RETPD), RLSE, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE keywords.
DIR(num)	The number of directory blocks for the output data set. If you do not enter DIR, the default is taken from the input data set.
SPACE(prim,sec)	The primary and secondary quantities for the output data set in track or cylinder units. If you do not enter BLK, TRK, or CYL but you enter SPACE, space allocation units are assumed the same as the current allocation.

BLK	Allocates the output data set in blocks. If you enter BLK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.
TRK	Allocates the output data set in tracks. If you enter TRK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.
CYL	Allocates the output data set in cylinder units. If you enter CYL without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.
ROUND	Used with BLK to specify that the data set allocation be rounded up to cylinder boundaries.
EXPDT(yyddd)	The expiration date for the data set. On and after the expiration date, delete or write over the data set. Use EXPDT(yyyyddd).
RETPD(nnnn)	Number of days (0 to 9999) for your data center to keep the data set.
RLSE	Release unused space in the data set after the copy.
STORCLAS(sc)	Storage class for the data set. Use this parameter to identify performance and availability requirements for data sets. Use it instead of UNIT and VOLUME keywords for SMS managed data sets to determine where the data set is allocated. To nullify this parameter, type **STORCLAS(*).**
MGMTCLAS(mc)	The management class for the data set. It establishes the migration, backup, and space release characteristics for SMS-managed data sets. To nullify this parameter, type **MGMTCLAS(*).**
DATACLAS(dc)	The data class for the data set. It provides data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL, and some VSAM SHR options for SMS-managed-data sets. To nullify this parameter, type DATACLAS(*). Use DATACLAS for data sets that are not SMS managed.
DSNTYPE(typ)	Specify typ as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS-managed data set.

Remarks

Use the COMBINE subcommand to consolidate members into a single data set or member. ISPF statistics are preserved, if possible, and IEBUPDTE controls in a member are converted to >< control statements.

COMBINE and SEPARATE are paired subcommands. To join members into a combined member or data set, use the COMBINE subcommand. To split the members apart again, use the SEPARATE subcommand or IEBUPDTE if the logical record length is 80.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
----------------- ISPMODE Session# 1 Log# 1 ------------- Row 531 to 541 of 541
COMMAND ===>                                               SCROLL ===> CSR
- DSN=WSER07.LIB.PDSE,VOL=SER=SER003  MEM=(#MDSECT  ------------------------
>------>combine * lib.cntl(download)
PDS290I COMBINE is in progress
PDS051I #MDSECT was combined; input=1,656; output=1,657
PDS051I @FILEXID was combined; input=5,852; output=5,853
PDS051I @FILEPAR was combined; input=2,264; output=2,265
PDS051I @SERVICE was combined; input=5,685; output=5,686
PDS051I @VERIFY was combined; input=1,953; output=1,954
PDS051I PDS0520 was combined; input=472; output=473

PDS054I Totals -- Members=6; Input=17,882; Characters=1,430,560

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

COMPARE Subcommand

The COMPARE subcommand displays differences between two members. Use the TSO COMPARE command. Add COMPARE operands after the member names.

If the current data set is not partitioned, code an ***** in the data set name position to compare the current data set to a different data set.

To compare directory entries between two libraries, use the COMPDIR subcommand. See the "COMPDIR Subcommand".

Example
```
COMPARE    oldmem   other.lib(newmem)
```

Syntax
```
COMPARE    oldmemb newmemb [DELTA       ]
                           [OVSUM       ]
                           [CHNG        ]
                           [SEQ|NOSEQ   ]
                           [COBOL       ]
                           [OVOL(volname)]
                           [NVOL(volname)]

                           Note: these parameters are for SuperC.
```

Aliases
```
CO, COM, COMP, COMPA, COMPAR, COMPARE
```

Defaults

DELTA

Required

oldmemb, newmemb

Operands

oldmemb	Base member to be compared with newmemb. *oldmemb* can be a data set and member name if a member from a different data set is used.
newmemb	New member to be compared with oldmemb. *newmemb* can be a data set and member name if a member from a different data set is used.

If you code this operand as the name of a partitioned data set with no member name, the base member name is used. Use this to compare two different levels of the same module. |
| other operands | Optional. May include any COMPARE operands. |

Remarks

Use the TSO COMPAREW (preprocessor to COMPAREX), COMPAREC (pre-processor to SuperC), or COMPARE$ (the Yale compare command). This choice is made during StarTool FDM installation. To determine which program is used, type a CONTROL DEFAULT subcommand and look for an output line beginning COMPARE calls.

This interface is optional. Use it if COMPAREX, SuperC, or the Yale compare program is installed at your installation.

COMPDIR Subcommand

The COMPDIR subcommand creates a sublist based on differences or similarities between data sets.

Example

```
COMPDIR   pep*  file(ispplib)
DEL *
```

Syntax

```
COMPDIR  memgroup
        {/ or dsname [VOLUME(volser)] | * | FILE(ddname) [NUMBER(numb)]}
         [ALIAS|NOALIAS                 ]
         [AS(pname) | TO(pname)         ]
         [SUBSTITUTE(oldstring|newstring)]
         [SEQ|NOSEQ                     ]
```

```
[EXIST|NOEXIST|CHANGED|NOCHANGED|DIRCHANGE|NODIRCHANGE]
[MEMBERS|MEMLIST|ML|NEWML|SUBLIST]
[LKEDDATE|NOLKEDDATE]
```

Aliases

COMPD, COMPDI, COMPDIR

Defaults

memgroup, NOALIAS, EXIST, SUBLIST, NOLKEDDATE

Required

memgroup, dsname, or * or FILE(ddname)

Operands

memgroup	Group of members to be compared to those in the other data set.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
dsname	Comparison data set. Use standard TSO dsname syntax. If you enter ***** in the dsname field, the previously used data set name used for a COPY or COMPDIR subcommand is used.
	As an alternative to the AS or TO keywords, enter partial member names in parentheses following the data set name, as in this example: the.dsn(pname)
VOLUME(volser)	For uncataloged data sets, identifies the output data set volume name.
FILE(ddname)	The output data set is already allocated with the provided ddname. Only disk data sets are supported.
NUMBER(numb)	Concatenation number for the DDNAME identified. *numb* defaults to 1. If *numb* is larger than the number of concatenated data sets, the last data set in the concatenation sequence is used.
ALIAS	Adds associated members (aliases, apparent aliases, and associated main members) to the list of names to check.
NOALIAS	Adds only specifically identified member names to the sublist of members.
AS(pname)	*pname* defines the name of any comparison members. For example, with AS(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are compared. AS and TO are equivalent keywords.
	If only a single member is entered and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.
TO(pname)	*pname* defines the name of any comparison members. For example, with TO(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are compared. TO and AS are equivalent keywords.
	If only a single member is entered and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.
SUBSTITUTE(old,new)	Used with a forward slash (/) to represent the current data set name. Use SUBSTITUTE to replace a string of characters in the source data set name to create a target data set name.
	For example, if the current data set is PDSEXXX.MSGS, the following two subcommands are equivalent:
	``` COMPDIR ABC* PDSEYYYY.MSGS COMPDIR ABC* / SUBSTITUTE(XXX/YYYY) ```
EXIST	Searches for members from the current member group that are also present in the compare data set.

NOEXIST	Searches for members from the current member group that are not present in the compare data set.
SEQ	Can use with **CHANGED** or **NOCHANGED** to ignore sequence numbers in source member compares.
NOSEQ	Default. Can use with **CHANGED** or **NOCHANGED** to compare sequence numbers in source member compares.
CHANGED	Searches for members from the current member group that are not present in the compare data set or those members that do not contain identical data. Actual member data is compared for this operand and if the members are in load data sets, you can get unequal compares if CSECTS are re-blocked or re-ordered in load modules being compared.
NOCHANGED	Searches for members from the current member group that are present in the compare data set and whose members contain identical data. Actual member data is compared for this operand and if the members are in load data sets, you can get unequal compares if CSECTS are re-blocked or re-ordered in load modules being compared.
DIRCHANGE	Searches for members from the current member group that are not present in the compare data set or those members whose directory entry is not identical to the corresponding member in the current data set.
NODIRCHANGE	Searches for members from the current member group that are present in the compare data set and whose directory entries are identical to the corresponding member in the current data set.
MEMBERS	Displays the names of members that satisfy the COMPDIR search condition without changing the current member group.
MEMLIST	Same as ML. Specifies that any member selected by COMPDIR displays in a MEMLIST display. If no members are selected, a null sublist is the result of the comparison.
ML	Same as MEMLIST. Specifies that any member selected by COMPDIR displays in a MEMLIST display. If no members are selected, a null supplest is the result of the comparison.
NEWML	Same as MEMLIST and ML except that the current MEMLIST is reset.
SUBLIST	Specifies that any members displayed by the COMPDIR subcommand are selected for inclusion in a new SUBLIST. If no members are selected, a null sublist is the result of the comparison.
LKEDDATE	Compares module link edit dates. Applies only to load libraries.
NOLKEDDATE	Does not compare module link edit dates. Applies only to load libraries.

Remarks

The COMPDIR subcommand compares two libraries that contain members with a known relationship (for example, a library containing a new level of a software product or a new 'SYS1.PARMLIB' for a system in test mode).

If the result of a COMPDIR subcommand is a sublist of members, the next subcommand should be one that performs an operation on all of the members in the sublist.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
---------------- ISPMODE Session# 1 Log# -----------  ROW 990 TO 1,007 OF 1,033
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CLIST,VOL=SER=SER002  MEM=(COMPR  -------------------
>----->compdir : lib.clistv   noexist
PDS175I The member names have been compared
PDS176I 63 members initially; 33 members dropped

PDS165I Members are: COMPR, DSP, MIGR8, NOMG8DSN, NOMIGR8, NULL, NULL2,
         PCLIST1, PDSABEND, PDSCALL, PDSCLIS, PDSN, PDSNN, PDSSE, PDSTR, QW,
         QWO, SCPASMCL, SPIFFY, SPZAP, SSCT, SSUS, SSVT, SYNTAX, TESTCLS,
         TESTJUMP, TESTPDST, TESTPRIM, TRYREX, VOLUME

PDS193I This group contains 30 members
>----->at
PDS230I ISPF Stats: VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
PDS230I  COMPR      01.00  00/05/07  00/05/07  8:31    31    31     0  SER07
PDS230I  DSP        01.03  02/12/19  02/10/08  9:07     2     2     1  SER07
PDS230I  MIGR8      01.02  93/05/21  93/05/21 11:07     9     4     9  #G146
PDS230I  NOMG8DSN
PDS230I  NOMIGR8
PDS230I  NULL       01.00  91/02/08  91/02/08 10:26     0     0     0  SER07
```

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
--------------- ISPMODE Session# 2 Log# 1 --------------  ROW 584 TO 595 OF 595
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.PDSE310,VOL=SER=SER006  MEM=(PDSZA@C  -----------------------
>----->compdir pdsz* lib.pdse221 changed
PDS175I The data has been compared
PDS176I 84 members initially; 47 members dropped

PDS165I Members are: PDSZA@C, PDSZA@P, PDSZABA, PDSZAID, PDSZALLO, PDSZALOC,
         PDSZAOF, PDSZATB, PDSZATTR, PDSZAUTH, PDSZAZA, PDSZCHAN, PDSZCOMA,
         PDSZCONT, PDSZCOPS, PDSZCOPY, PDSZCREA, PDSZDUP, PDSZEXCL, PDSZFIND,
         PDSZFINM, PDSZFIXE, PDSZFIXP, PDSZFSTR, PDSZHIST, PDSZINIT, PDSZINST,
         PDSZMAP, PDSZPDSE, PDSZPDST, PDSZPRIN, PDSZREPL, PDSZRST1, PDSZRST2,
         PDSZSVCM, PDSZVERI, PDSZXREF

PDS193I This group contains 37 members
****************************** BOTTOM OF DATA ******************************
```

```
  FUNCTIONS  CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
--------------- Source MEMLIST 1, Session# 1  --------------- Row 1 to 25 of 37
 COMMAND ===>                                            SCROLL ===> CSR
- DSN=SER07.LIB.PDSE310,VOL=SER=SER006  MEM=(PDSZA@C  -----------------------
 CMD  NAME      DATA/MSG VER.MOD  CREATED    LAST  MODIFIED   SIZE  INIT  ID
      PDSZA@C
      PDSZA@P
      PDSZABA
      PDSZAID
      PDSZALLO
      PDSZALOC
      PDSZAOF
      PDSZATB
      PDSZATTR
      PDSZAUTH
      PDSZAZA
      PDSZCHAN
      PDSZCOMA
      PDSZCONT
```

COMPRESS Subcommand

The COMPRESS subcommand invokes IEBCOPY (or a compatible program) to compress a partitioned data set.

NOTE If a COMPRESS operation does not complete because of a software or hardware error, your data set may be destroyed since data sets are updated in place.

For a PDSE data set, the COMPRESS subcommand invokes the COPY subcommand to reorganize the data set as follows:

1   If you have proper access authority for the source data set, a COPY subcommand is issued with the COPYBACK keyword.

2   A new temporary PDSE similar to the source data set is allocated.

3   All members are copied into the new temporary target data set.

4   The source PDSE is reset to NEW status (all members are deleted).

5   All members are copied back into the source data set.

6   If TRIM is requested, all unused tracks are released.

Example
```
COMPRESS   summary
```


Syntax
```
COMPRESS   [SUMMARY | LIST | NOLIST   TRIM]
```

Aliases

COMPR, COMPRE, COMPRES, COMPRESS

Defaults

SUMMARY

Required

None

Operands

SUMMARY	Displays only compress summary and error messages.
LIST	Displays all compress messages.
NOLIST	Displays no compress messages.
TRIM	After the compress operation, trim the data set as follows: ■ For a PDS data set, release any unused extents as for FIXPDS RELEXT ■ For a PDSE data set, release all unused tracks as for FIXPDS RELEASE

Remarks

You may have chosen any one of several interfaces during StarTool FDM installation. To determine which interface you selected, type a CONTROL DEFAULT subcommand and look for an output line beginning COMPRESS calls. Any of the following are supported:

IEBCOPY	Invokes IEBCOPY (or PDSFAST if IEBCOPY is renamed) directly.
PDSFAST	Invokes PDSFAST directly.
IKJEFTSR	Uses the TSO/E parallel terminal monitor program interface to authorized tasks. **NOTE** Using interface causes a session lock-up if you invoke PDSFAST and use SHR allocation.
anyother	Uses the TSO/E parallel terminal monitor program interface with potential lock-up problems if you invoke PDSFAST.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-----------------------------------------------------------------------------
------------------------- ISPMODE Session Display ------------  ROW 14 OF 19
COMMAND ===>                                               SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=(CPX623 -----------------------
>----->compress
IEB144I  THERE ARE 0000124 UNUSED TRACKS IN OUTPUT DATA SET
IEB149I  THERE ARE 0000025  UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I  END OF JOB -00 WAS HIGHEST SEVERITY CODE
```

```
---------------- ISPMODE Session# 1 Log# 1 ---------------- ROW 28 TO 47 OF 63
 COMMAND ===>                                          SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.SMSLIB.PDSEXTN2,VOL=SER=STG003  MEM=:  -----------------------
 >------> C SMSLIB.PDSEXTN2
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390 C   FB       80  32720  26X      63        1     1 TRK NOLIMIT

 PDS300A ENTER OPTION -- DSN=WSER07.SMSLIB.PDSEXTN2,VOL=SER=STG003  MEM=CUS420
 >----->compress
 PDS041I this data set is a PDSE; it will be reorganized


 Copy    : '&&TEMP' NEW NOBLDL NODUP COPYBACK SUMMARY
 PDS174I '&&TEMP' has been created on volume STG009
 IEB1135I IEBCOPY  FOR JDZ1110  SERVICE LEVEL UY97739  DATED 08/26/93
 IEB1035I WSER07    $SERPDSE $SERPDSE 06:33:21 WED 02 AUG 1995 PARM='' A0
 IEB1064I   STANDARD DD NAMES- SYSIN    SYSPRINT SYSUT1   SYSUT2   SYSUT3   SYS
 IEB1065I OVERRIDING DD NAMES- SYS00011 SYS00013 SYS00008 SYS00010 SYS00015 SY
 IEB1057I VL GETMAIN REQUESTED 250K TO 1M BYTES.  OBTAINED 1M. WILL RELEASE 8K
  COPY    O=SYS00010,LIST=NO,I=SYS00008
 IEB1058I ALLOCATED 2 CONTIGUOUS BUFFERS EACH 111K BYTES.  WORK AREA HAS 790K
 IEB1013I COPYING FROM PDSE  INDD=SYS00008 VOL=STG003 DSN=WSER07.SMSLIB.PDSEXT
 IEB1014I           TO PDSE OUTDD=SYS00010 VOL=STG009 DSN=SYS95214.T063320.RA0
 IGW01550I 154 OF 154  MEMBERS WERE COPIED
 IEB1016I RETURN CODE IS 0
 IEB1056I RELEASED 1016K ADDITIONAL BYTES.
 IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE

 IEB1135I IEBCOPY  FOR JDZ1110  SERVICE LEVEL UY97739  DATED 08/26/93
 IEB1035I WSER07    $SERPDSE $SERPDSE 06:33:27 WED 02 AUG 1995 PARM='' A0
 IEB1064I   STANDARD DD NAMES- SYSIN    SYSPRINT SYSUT1   SYSUT2   SYSUT3   SYS
 IEB1065I OVERRIDING DD NAMES- SYS00011 SYS00013 SYS00010 SYS00008 SYS00015 SY
 IEB1057I VL GETMAIN REQUESTED 250K TO 1M BYTES.  OBTAINED 1M. WILL RELEASE 8K
  COPY    O=SYS00008,LIST=NO,I=SYS00010
 IEB1058I ALLOCATED 2 CONTIGUOUS BUFFERS EACH 111K BYTES.  WORK AREA HAS 790K
 IEB1013I COPYING FROM PDSE  INDD=SYS00010 VOL=STG009 DSN=SYS95214.T063320.RA0
 IEB1014I           TO PDSE OUTDD=SYS00008 VOL=STG003 DSN=WSER07.SMSLIB.PDSEXT
 IGW01550I 154 OF 154  MEMBERS WERE COPIED
 IEB1016I RETURN CODE IS 0
 IEB1056I RELEASED 1016K ADDITIONAL BYTES.
 IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390 C   FB       80  32720  26X      63       33     1 TRK NOLIMIT
```

CONDEND Subcommand

Use the CONDEND subcommand to terminate StarTool FDM if the previous subcommand
has one or more warning or error level messages. The return code is set to the numeric
value of the message that terminated StarTool FDM. After StarTool FDM terminates by
CONDEND, the return code is in the range of 400 to 999; however, if RC is entered, it is
translated to 4 or 8.

Example

```
CONDEND exit(450)
```

Syntax

```
CONDEND  [WARNINGS|ERRORS|EXIT(n1,n2, ...)|NOEXIT(n1,n2, ...)]
         RC
```

Aliases

```
COND, CONDE, CONDEND
```

Defaults

WARNINGS

Required

None

Operands

WARNINGS	Exit if any message issued by the last subcommand was in the range PDS400W through PDS999E (warning or error messages).
ERRORS	Exit if any message issued by the last subcommand was in the range PDS600E through PDS999E (error messages).
EXIT	Exit if a named warning or error message was issued by the last subcommand. For example, to exit for a PDS450W, PDS715E, or PDS823E message, type CONDEND EXIT(450,715,823)
NOEXIT	Exit for any warning or error message except for those messages in an exemption list. For example, to exit for any warning or error message except PDS460W, PDS958E or PDS609E, type CONDEND NOEXIT(460,958,609)
RC	Translates the return code to four if the terminating message was between PDS400W through PDS599W or eight if the terminating message was between PDS600E and PDS999E.

Remarks

Use this subcommand to check that a critical subcommand in a CLIST is working as expected. StarTool FDM CLIST continues to execute after receiving program warning or error messages. With this facility, terminate a StarTool FDM CLIST process when it is not functioning properly. You can detect this problem from a driving CLIST by examining &LASTCC.

Put CONDEND statements in your CLIST just after any critical subcommands.

CONTROL Subcommand

The CONTROL subcommand specifies global and immediate processing options.

Example

```
CONTROL   DSNAME(lib.cntl(pdsdata)) OLD
```

Syntax

```
CONTROL   [ALIASINFO|NOALIASINFO]
          [LKEDDATE|NOLKEDDATE]
          [PROMPT|NOPROMPT]
          [RECOVER|NORECOVER]
          [TRANSLATOR|NOTRANSLATOR]
          [SINGLE|DOUBLE|MULTIPLE|RETAIN(ntrk)]
          [INFO|NOINFO]
          [PUTLINE|NOPUTLINE]
          [COMMAND|NOCOMMAND]

          [SYSOUT(c)
             {FORM(ffff)|NOFORM|DEST(destination)|NODEST}|
             DSNAME(memb) {OLD|MOD|NEW|SHR}|
             NODSN|
             NOSYSOUT]

          [DEFAULTS|IOSTATS|LISTENV|RESTRICTED|TESTABEND|TESTMSGS|
           TESTOUTLOOP|TESTPARSE|TESTREAD|TESTSYNTAX]
```

Aliases

```
CON, CONT, CONTR, CONTRO
```

Defaults

Initially in TSO: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR, RETAIN(9)
Initially in the background: ALIASINFO, LKEDDATE, NOPROMPT, NORECOVER,
TRANSLATOR, RETAIN(9)

Required

None

Operands

ALIASINFO	Provides alias information for ATTRIB subcommands. Also provides alias information for MAP subcommands with alias members if MAP has no CSECT-limiting parameters.
NOALIASINFO	Provides no alias or main member identification for ATTRIB and MAP subcommands.
LKEDDATE	The date of the linkage edit on ATTRIB subcommands.
NOLKEDDATE	Does not provide the date of the linkage edit on ATTRIB subcommands.
PROMPT	Solicits YES/NO responses from the terminal operator unless StarTool FDM is executing in the background, from CLIST input or from a storage stack. YES/NO prompting occurs at the following decision points: ■ RENAME - before renaming a group of members ■ DELETE - before deleting a default member. Avoid this by tying DELETE * to delete the default member (a single member) ■ DELETE - before deleting a list or a group of members ■ FIXPDS - before modifying a data set ■ SUBMIT - before submitting a group of members ■ RESTORE - before restoring a member found in scan mode. A CAN reply is also acceptable
NOPROMPT	Does not solicit YES/NO responses; assumes a YES response in each case.
RECOVER	Attempts ESTAE recovery after a program ABEND.
NORECOVER	Does not attempt ESTAE recovery after a program ABEND. Subcommands that specifically test for ABEND conditions (such as IF with LOADERR or VERIFY with LOAD) recover from ABEND situations regardless of RECOVER/NORECOVER.
TRANSLATOR	Outputs assembler/compiler TRANSLATOR information from HISTORY subcommands.
NOTRANSLATOR	Does not output assembler/compiler TRANSLATOR information from HISTORY subcommands.
SINGLE	Uses single buffering for input operations. This operand is ignored for PDSE data sets.
DOUBLE	Uses double buffering for input operations wherever possible. This operand is ignored for PDSE data sets.
MULTIPLE	Uses multiple buffering for input operations wherever possible. This mode reads an entire disk track with each input operation. This operand is ignored for PDSE data sets.

RETAIN(*ntrk***)**	The number of disk track images (one through nine) to keep in storage buffers. Each new member read operation searches these buffers before performing a read multiple EXCP operation. EXCP operations are saved during the execution of a single subcommand. Each additional buffer requires about 58K of storage. If you later enter RETAIN(n) with a lower number, extra buffers are *not* returned to the system. A default number of RETAIN buffers are allocated during StarTool FDM initialization.
SYSOUT(c)	Starts a session log output to the SYSOUT class specified. Enter DEST and FORM keywords as additional SYSOUT specifications. The SYSOUT output continues until the end of the program or until you enter CONTROL SYSOUT / NOSYSOUT / DSNAME / NODSN.
NOSYSOUT	Terminates any SYSOUT or DSNAME session log output.
DEST(*destination***)**	The destination for SYSOUT data (used only with the SYSOUT keyword). Enter *dst* as a 1- to 8-character JES destination or a name of the form nodename.userid
NODEST	Uses the default destination for routing SYSOUT data.
FORM(ffff)	The forms name for SYSOUT (used only with the SYSOUT keyword). Enter *ffff* as a 1- to 4-character JES form name.
NOFORM	Uses the default forms name for SYSOUT data.
DSNAME(*member***)**	Starts a session log output to the named data set. The DSNAME output continues until the end of the program or until you enter CONTROL SYSOUT / NOSYSOUT / DSNAME / NODSN. In addition: ■ Enter a member name for session log output to a partitioned data set ■ Enter a data set password ■ Enter a data set disposition (OLD|MOD|SHR|NEW) ■ If you enter a data set disposition of NEW, the output data set is created by the CONTROL subcommand with space parameters as follows: SPACE=(TRK,(1,4))
NODSN	Terminates any SYSOUT or DSNAME session log output.
OLD	Obtains exclusive use on the session log data set; it may not be shared with other users.
MOD	Adds to the end of any existing session log output data set. If the data set does not exist, it is created with DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120), SPACE=(TRK,(1,4))
SHR	Shares the session log data set with other users. Since the data set is modified, minimize the use of this parameter.
NEW	Creates a session log data set with DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120), SPACE=(TRK,(1,4))
INFO	Resumes output of StarTool FDM informational messages.
NOINFO	Suppresses StarTool FDM informational messages. This is intended for use in background sessions or in any situation where the normal informational messages are not required.

PUTLINE	Resumes output of PUTLINE data from StarTool FDM.
NOPUTLINE	Suppresses StarTool FDM PUTLINE output (except error messages). This is intended for use in CLISTs where the amount of StarTool FDM output is to be minimized.
COMMAND	Resumes log recording of StarTool FDM commands.
NOCOMMAND	Suppresses log recording of StarTool FDM commands. This is intended for use when SYSOUT/DSN recording is being performed and recording of commands and prompting messages is not wanted.
DEFAULTS	Displays a list of StarTool FDM defaults set by the installation default module. This list includes external programs called, global defaults, and a security summary.
IOSTATS	Displays statistics relating to input buffering. Only non-zero quantities display and statistics are set to zero after each display.
LISTENV	Displays information on the hardware and software environment. This includes information on the last IPL and the system catalog.
RESTRICTED	Displays a list of restricted use subcommand names. Users can restrict subcommands at StarTool FDM installation. This message lists subcommand and operand combinations that your installation chose to restrict. This operand may be not be available for your use since it can also be restricted.
TESTABEND	Provides a S0C1 ABEND (or some other ABEND) for program testing.
TESTMSGS	Displays all StarTool FDM numbered messages for program testing.
TESTOUTLOOP	Enters an intentional output message loop for program testing.
TESTPARSE	Switches the parse processor. When CONTROL TESTPARSE is first issued, the StarTool FDM parser is disabled in favor of IKJPARS. The next use of TESTPARSE toggles the StarTool FDM parser back again. When the StarTool FDM parser is disabled, a global message with NOPARSE generates as feedback. This facility is for program testing.

TESTREAD	Tests the input EXCP routine using an initial TTR of 000001 for each of the following cases:
	■ BPAM - reads using a BPAM DCB for PDSE data sets or a BSAM DCB for striped data sets. Other tests are not performed for these data sets. The return code is 00 and NEXT TTR ADDRESS is 000001
	■ SINGLE - reads single (each read obtains one physical block). The return code is 00 and NEXT TTR ADDRESS is 000002 if sufficient data is available
	■ DOUBLE - reads double (each read obtains two physical blocks). The return code is 00 and NEXT TTR ADDRESS is 000003 if sufficient data is available
	■ MULTIPLE - reads multiple (each read obtains an entire track). If the disk unit supports the READ MULTIPLE CCW, the return code is 00 and NEXT TTR ADDRESS is 000101 if sufficient data is available
	If the disk unit does not support the READ MULTIPLE CCW, you get a PDS892E message. The return code is 00, and NEXT TTR ADDRESS is 000003 (double buffering is used).
TESTSYNTAX	Tests subcommand syntax only. After CONTROL TESTSYNTAX is issued, StarTool FDM subcommands are syntax checked (except for CHANGE, CONTROL, END, GO, QUIT, and ISPMODE commands). TESTSYNTAX toggles StarTool FDM subcommands again. When the StarTool FDM subcommand syntax is disabled, a global message with TESTSYNTAX generates as feedback. This facility tests the syntax of subcommands before using them in another process.

Remarks

Most CONTROL subcommand operands (such as ALIASINFO, SINGLE, and TESTPARSE) are global in that they remain in effect until reset by another CONTROL subcommand. The other operands (such as TESTABEND and IOSTATS) are immediate and take effect for only a single CONTROL subcommand.

CONTROL does not use ISPF services; however, if StarTool FDM is invoked from READY mode, StarTool FDM attempts to establish an ISPF environment for the first CONTROL subcommand containing a SYSOUT or DSN keyword so that subsequent subcommands can use ISPF services.

The input buffering type (BPAM, SINGLE, DOUBLE, MULTIPLE, or RETAIN) is reset for each data set allocated according to the data set organization or the device type on which the data set resides. RETAIN buffering is used for device types that support the READ MULTIPLE CCW. DOUBLE is used otherwise. BPAM mode is selected automatically for PDSE data sets and cannot be selected explicitly.

```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
--------------------- ISPMODE Session Display --------------- ROW 447 OF 455
COMMAND ===>                                            SCROLL ===> CSR
- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=:  ------------------------------
>----->control retain(9)


PDS100I STARTOOL/SuperEdit -- Version 6.2.0  2000.001

PDS030I Global operands: NOTRANSLATOR, ALIASINFO, LKEDDATE, PROMPT, RECOVER
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(9)
PDS036I Largest free storage area is 2364K
PDS046I Largest area above the line is 1905M
***************************** BOTTOM OF DATA ******************************
```

```
------------------------- ISPMODE Session# 1 Log# 1 ----- ROW 26 TO 44 OF 95
COMMAND ===>                                            SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR849  MEM= ---------------------------------
>----->cont defaults


PDS100I STARTOOL/SuperEdit -- Version 6.2.0  2000.001

PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(9)
PDS036I Largest free storage area is 1304K
PDS046I Largest area above the line is 1895M

PDS037I Installation defaults from PDS#OPT4 1999/06/08 07.34:
 Access control method            NONE
 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
 S line command - LISTC           C
 BLK3390  calls                   BLK3390
 BLK9345  calls                   BLK9345
 BROWSE   calls                   PBROWSE
 COMPARE  calls                   COMPAREC
 COMPRESS calls                   IKJEFTSR
 COPY     calls                   IKJEFTSR
 EDIT     calls                   PEDIT
```

```
------------------------ ISPMODE Session# 1 Log# 1 -----  ROW 33 TO 52 OF 77
 COMMAND ===>                                              SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.PRINT,VOL=SER=STR961  ----------------------------------------
 >----->cont listenv


 PDS100I PDSE/SuperEdit -- Version 6.2.0  2000.001


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


 PDS280I System serial:05905; CPU type:5995
 PDS281I Active CPUs:4 5 6 7
 PDS282I SMF ID:SCU1; System mode:ESA/370
 PDS283I Maintenance data:SP6.0.2 JBB6602
 PDS284I IPL date:97/04/14 97.104; 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:00


 PDS052I Real storage is 472M; expanded 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  090FFFFF     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  007FFFFF      8,172K
 PDS059I V=R AREA           00005000  00024FFF        128K
 PDS059I SYSTEM             00001000  00004FFF         16K
 PDS059I PSA                00000000  00000FFF          4K
 ****************************** BOTTOM OF DATA ***************************
```

NOTE Be aware that values displayed for real and extended storage in message `PDS052` on the CONTROL LISTENV panel may not be accurate when `VARY STOR` commands are in progress.

COPY Subcommand

The COPY subcommand builds copy control statements and invokes a copy program to selectively copy members to another data set.

Example

```
COPY  mem1:mem2  other.dataset  new  tra  spa(5,5)  dir(20)
```

Syntax

```
COPY  memgroup
      {/ or dsname [VOLUME(volser)] | * | FILE(ddname) [NUMBER(numb)]}
      [OLD|SHR | NEW [BLK|TRK|CYL]      [DIR(num)]  [SPACE sec)]]
                     [ROUND]  [EXPDT(yyddd)|RETPD(nnnn)]  [RLSE]
                     [STORCLAS(sclass)]  [MGMTCLAS(mclass)]
                     [DATACLAS(dclass)]  [DSNTYPE(LIBRARY|PDS)]
                     [UNCTLG|NOCATLG]
      [ALIAS|NOALIAS                ]
      [AS(pname) | TO(pname)        ]
      [CALL|NOCALL                  ]
      [COPYBACK                     ]
      [EXIST|NOEXIST|EXCLUDE|NOBLDL]
      [MAXBLK(num)                  ]
      [REPLACE|NOREPLACE            ]
      [SUMMARY|LIST|NOLIST          ]
      [UNIT(Unitname)               ]
      [CHANGE|GO|TRIM               ]
      [NODUP                        ]
      [ADDZAP|ADDZAP(number)        ]
      [NONUM|SNUM                   ]
      [CLIST|NOCLIST                ]
      [REXX                         ]
      [MOVE                         ]
      [SUBSTITUTE(oldstring|newstring)]
    * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)  ]
    * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)         ]

*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases

```
COP, COPY
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

ALIAS if LOAD, NOALIAS otherwise; SHR, CALL, NOREPLACE, SUMMARY; if NEW is specified, BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE default to values used for the input data set.

Required

memgroup, dsname or * or FILE(ddname)

Operands

memgroup	The members to be copied to the target data set. Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
dsname	The output data set. Use TSO dsname syntax or a temporary data set name such as '&&TEMP'. If you enter ***** in the dsname field, the previously used data set name (for a COPY command) is used instead. As an alternative to the AS or TO keywords, enter partial member names in parentheses following the data set name, as in this example: **the.dsn(pname)**
VOLUME(volser)	For NEW data sets, specifies the output volume name. For existing data sets, use this parameter if the output data set is not cataloged. For new data sets, code VOLUME(*) to mean the volume parameter is to be copied.
FILE(ddname)	The output data set is already allocated with the provided ddname. Only disk data sets are supported. If you use the FILE format, then do not use dsname, OLD, SHR, NEW, RLSE, and data set allocation keywords. If the data set is concatenated, use the SHR or OLD keyword since StarTool FDM reallocates the output data set.
NUMBER(numb)	The concatenation number for the DDNAME identified. *numb* defaults to 1. If *numb* is larger than the number of concatenated data sets, the last data set in the concatenation sequence is used.
OLD	Obtains exclusive use of the output data set.
SHR	Allows shared use of the output data set. StarTool FDM provides output data set protection. See Appendix B, "Update Protection".
NEW	Creates the output data set. When you specify NEW, also specify BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT (or RETPD), RLSE, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE keywords.
DIR(num)	The number of directory blocks for the output data set. If you do not enter DIR, the default is taken from the input data set.
SPACE(prim,sec)	If you do not enter BLK, TRK, or CYL but you enter SPACE, the space allocation units are assumed the same as the current allocation.
BLK	Allocates the output data set in blocks. If you enter BLK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.
TRK	Allocates the output data set in tracks. If you enter TRK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.

CYL	Allocates the output data set in cylinder units. If you enter CYL without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.
ROUND	Used with BLK to specify that the data set allocation be rounded up to cylinder boundaries.
EXPDT(yyddd)	The expiration date for the data set. On and after the expiration date, delete or write over the data set. Also use the format EXPDT(yyyyddd).
RETPD(nnnn)	The number of days (0 to 9999) for your data center to retain the data set.
RLSE	Any unused space in the data set be released after the copy operation.
STORCLAS(sc)	The storage class for the data set. Use this parameter to identify performance and availability requirements for data sets. Use this parameter instead of UNIT and VOLUME keywords for SMS managed data sets to determine where the data set is allocated. To nullify this parameter, type **STORCLAS(*)**
MGMTCLAS(mc)	The management class for the data set. This parameter establishes the migration, backup, and space release characteristics for SMS managed data sets. To nullify this parameter, type **MGMTCLAS(*)**
DATACLAS(dc)	The data class for the data set. Use this parameter to provide data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL and some VSAM SHR options for SMS managed data sets. To nullify this parameter, type **DATACLAS(*)** **Use DATACLAS for data sets that are not SMS managed.**
DSNTYPE(typ)	Specify *typ* as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS managed data set.
UNCATLG	Does not catalog the NEW data set.
NOCATLG	Does not catalog the NEW data set.
ALIAS	Copies any associated members (aliases, apparent aliases, and associated main members).
NOALIAS	Copies only the members specifically identified.
AS(pname)	*pname* defines the name of any output members. For example, with AS(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are created. AS and TO are equivalent keywords. If only a single member is input and NOALIAS is in effect, the resulting member name is taken only from the AS or TO keyword.
TO(pname)	*pname* defines the name of any output members. For example, with TO(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 is created. TO and AS are equivalent keywords. If only a single member is input and NOALIAS is in effect, the resulting member name is taken only from the AS or TO keyword.

CALL	Calls the copy program to perform the copy.
NOCALL	Does not call the copy program. Member checking, control card generation, and data set allocations are still performed.
COPYBACK	Used by the COMPRESS subcommand to reorganize PDSE data sets with the COPY subcommand. After all data is copied to the target data set, the source data set is reset to NEW status (all members are deleted) and all members are copied back into the source data set again.
EXIST	Copies only the subset of memgroup that is also present in the output data set (the REPLACE keyword is also used).
NOEXIST	Copies only the subset of memgroup that is ***not*** present in the output data set.
EXCLUDE	Inverts the normal copy "select" statements to "exclude" statements. This option also turns off BLDL (member existence) checking for the target library.
NOBLDL	Does not perform BLDL (member existence) checking in the target library. Use this parameter when performance of the COPY subcommand is very important and BLDL checking (by StarTool FDM) for possible errors is not important.
MAXBLK(num)	The maximum member block size. For load members, a COPYMOD operation is invoked for this keyword or if the input and output data set block sizes differ.
REPLACE	Replaces members in the output data set by any copied members.
NOREPLACE	Does not replace members in the output data set by copied members. Warning messages are issued to document any uncopied member names unless you enter EXIST, NOEXIST, EXCLUDE, or NOBLDL.
SUMMARY	Displays only copy summary and error messages.
LIST	Displays all copy messages.
NOLIST	Displays no copy messages.
UNIT(Uname)	Specifies the allocation unit name. Use this parameter if the default unit name is incorrect for this allocation.
CHANGE	StarTool FDM should CHANGE to the output data set at the end of the COPY process.
GO	StarTool FDM should initiate a GO session for the output data set at the end of the COPY process.
TRIM	After the copy operation, trim the source data set. ■ For a PDS data set, release any unused extents as for FIXPDS RELEXT ■ For a PDSE data set, release all unused tracks as for FIXPDS RELEASE
NODUP	Does not use internal copy routines. Call an external copy utility.

ADDZAP	For load members only, specifies that a ZAP IDR record is to be added (each ZAP IDR record contains space for 19 historical entries).
ADDZAP(n)	For load members only, specifies that one or more ZAP IDR records are to be added (each ZAP IDR record contains space for 19 historical entries).
NONUM	For source members only, specifies that the input members are all data and no sequence numbers are to be formatted. This keyword is ignored on a RECFM=F to RECFM=V CLIST conversion.
SNUM	For source members only, specifies that sequence numbers in the input (the first eight columns for RECFM=V data sets; otherwise the last 8 columns) are to be discarded during the copy.
NOCLIST	Do not perform CLIST conversion. CLIST conversion is performed if RECFM=V, LRECL=255 data is copied to RECFM=F, LRECL=80 or the reverse. CLIST conversion entails assigning a new sequence number and breaking up long lines into several short ones while maintaining normal CLIST format rules.
CLIST	Invokes an internal copy routine and converts CLISTs if required.
REXX	Specifies that RECFM=VB, LRECL=255 REXX members be converted to RECFM=FB, LRECL=80 REXX members using REXX continuation rules.
MOVE	Specifies that if no errors are detected during the copy process, delete all of the copied members.

SUBSTITUTE (old,new)	Used with a forward slash (/) to represent the current data set name. Use SUBSTITUTE to replace a string of characters in the source data set name to create a target data set name.
	 For example, if the current data set is PDSEXXX.MSGS, the following two subcommands are equivalent:
	 `COPY ABC* PDSEYYYY.MSGS`
	 `COPY ABC* / SUBSTITUTE(XXX/YYYY)`

FROMKEY(ky)	For VSAM data sets only, code ky as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Can be used with TOKEY. It can be specified only for an alternate index or a key-sequenced data set.

FROMADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS.
	FROMADDRESS(address):
	■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components
	■ Cannot be specified if the data set is being accessed through a path
	■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.
	Use this parameter with TONUMBER. It can be specified only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, code ky as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified.
	Use this parameter with FROMKEY. It can be specified only for an alternate index or a key-sequenced data set.
TOADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS.
	TOADDRESS(address):
	■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components
	■ Cannot be specified if the data set is being accessed through a path
	■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.
	Use this parameter with FROMNUMBER. It can be specified only for a variable or fixed relative record data set.

Required

memgroup, dsname or * or FILE(ddname)

Remarks

The COPY/DUP code switches dynamically to StarTool FDM internal copy routines if requested for a given subcommand or if you require DUP internal support. COPY/DUP switches to an internal copy routine for any of the following situations:

- The DUP subcommand name initiates a copy

- The input data set is VSAM or not partitioned

- The output data set is not partitioned

- An ADDZAP, CLIST, NOCLIST, NONUM, or SNUM keyword is used

- Source data set and the MAXBLK keyword is used

- Conversion of RECFM=FB to RECFM=VB or RECFM=VB to RECFM=FB.

- Output records requiring truncation

- RECFM=FB output records requiring blank padding at the end

- Members are copied back into the source data set

For a PDSE data set, invoke the COPY subcommand to support the COMPRESS subcommand using the COPYBACK keyword as follows:

- Creates a new temporary PDSE similar to the source data set

- Copies all members into the new temporary target data set

- Resets the source PDSE to NEW status; deletes all members

- Copies back all members into the source data set

IEBCOPY (or a compatible PDS copy program) performs the data copy operation. Internal StarTool FDM routines are used instead if the input data set is sequential or a single member is copied to a sequential output data set. In the following cases, an external copy program such as IEBCOPY or PDSFAST is required:

- Load library and the MAXBLK keyword is used

- Load library and the BLKSIZE is more than output BLKSIZE

- The new NODUP keyword is used

If the output data set is sequential and more than one input member is selected for copy, the PDS copy program creates an output data set in IEBCOPY unload format.

If you use the TO (or the equivalent AS) keyword, each member is renamed in the copy in a generic fashion. For example, if you enter TO(X12), each resulting member name is formed by placing X12 over the first three characters of the original member name. An exception to this rule is made when only a single member is to be copied and its aliases are not also requested. In this case, the member name is taken directly from the TO keyword.

If you do not specify the keywords EXIST, NOEXIST, EXCLUDE, or AS/TO and you enter COPY : to copy all members, the normal IEBCOPY select statements are not generated. However, if you want to build select statements for the entire library, specify a member group name on the COPY subcommand which includes all members such as COPY :9.

Request an IEBCOPY COPY operation; however, if the data set contains load modules (RECFM is U) and the input and output data set block sizes differ, a COPYMOD operation is requested with a MAXBLK operand set to the output data set block size. Explicitly request

a COPYMOD operation for load module libraries by specifying MAXBLK and the desired maximum member block size.

> **NOTE** This subcommand modifies the output data set. To ensure data set integrity, allocate the output data set as OLD or NEW, or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. See Appendix B, "Update Protection".

You may have chosen any one of several interfaces during StarTool FDM installation. To determine which interface you chose, type a CONTROL DEFAULT subcommand and look for an output line beginning COPY calls. For installations that use PDSFAST, also see the COPY utility for program objects. The following are supported:

IEBCOPY	Invokes IEBCOPY (or PDSFAST if IEBCOPY is renamed) directly.
PDSFAST	Invokes PDSFAST directly.
IKJEFTSR	Uses the TSO/E parallel terminal monitor program interface to authorized tasks. **NOTE** This interface causes a session lock-up if you invoke PDSFAST and use SHR allocation.
anyother	Uses the TSO/E parallel terminal monitor program interface with potential lock-up problems if you invoke PDSFAST.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session Display ------------- ROW 387 OF 429
COMMAND ===>                                                SCROLL ===> CSR
- DSN=C911407.LIB.PDS98TST,VOL=SER=STR811  MEM=:  ---------------------------
>----->copy :9 lib.test new
PDS174I 'C911407.LIB.TEST' has been created on volume STR801
         COPY    O=SYS00111,I=SYS00102
         S M=@DIACLIK,@DIAINIT,@DIALOG,@DIAPANL,@DIAPART,@DIASTAK,@DIATBL
         S M=@DIAWRK,ALIAS1,ALLGLOB,ASMEXT2,CHECKOUT,CICS,COMPARE,COMPARES
         S M=COMPA1,COMPA2,COMPA3,DAFABC,DAFJCLO,DISAMSM,DISAMXM,DISASME
         S M=DISASMX,DISASM3,DISDAFD,DISDAFE,MAPXREF,PDS99,PDS99T1D,TESTMSG
IEB167I  FOLLOWING MEMBER(S)  COPIED FROM INPUT DATA SET REFERENCED BY SYS0010
IEB154I  @DIACLIK HAS BEEN SUCCESSFULLY  COPIED
IEB154I  @DIAINIT HAS BEEN SUCCESSFULLY  COPIED
IEB154I  @DIALOG  HAS BEEN SUCCESSFULLY  COPIED
IEB154I  @DIAPANL HAS BEEN SUCCESSFULLY  COPIED
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
--------------------------- ISPMODE Session Display ----------- ROW 31 OF 48
COMMAND ===>                                             SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=#*  ----------------------------
>----->create link.test new tra spa(10,10) blk(6000)
PDS174I 'C911407.LINK.TEST' has been created on volume STR803
>----->copy * link.test
         COPYMOD O=SYS00032,MAXBLK=06000,I=SYS00022
         S M=#DISDF01,#DISXREF,#OPCODE,#OPCODEM,#OPCODE2,#PDSMAIN
IEB190I  MAXIMUM BLOCK SIZE IS     6000, MINIMUM BLOCK SIZE IS     1024
IEB167I  FOLLOWING MEMBER(S)  COPIED FROM INPUT DATA SET REFERENCED BY SYS0002
IEB19AI  MEMBER #DISDF01 COPIED AND REBLOCKED
IEB19AI  MEMBER #DISXREF COPIED AND REBLOCKED
IEB19AI  MEMBER #OPCODE  COPIED AND REBLOCKED
IEB19AI  MEMBER #OPCODEM COPIED AND REBLOCKED
IEB19AI  MEMBER #OPCODE2 COPIED AND REBLOCKED
IEB19AI  MEMBER #PDSMAIN COPIED AND REBLOCKED
IEB144I  THERE ARE 0000001 UNUSED TRACKS IN OUTPUT DATA SET
IEB149I  THERE ARE 0000079  UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I  END OF JOB -00 WAS HIGHEST SEVERITY CODE
```

CREATE Subcommand

The CREATE subcommand allocates a new data set based on the current data set.

MODEL is also available as a full-screen front-end to CREATE as shown in an example later.

Example
```
CREATE   new.data  space(10,10)
```

Syntax
```
CREATE / or dataset
      [BLKSIZE(Bval)                 ]
      [DIR(num)|NODIR                ]
      [LRECL(Lval)                   ]
      [OPTCD(W|WC|WCZ|C|CZ|Z)        ]
      [SPACE(nprim,nsec)             ]
      [BLK|TRK|CYL                   ]
      [ROUND                         ]
      [RETPD(nnnn)|EXPDT(yyddd)      ]
      [STORCLAS(sclass)             ]
      [MGMTCLAS(mclass)             ]
      [DATACLAS(dclass)             ]
      [DSNTYPE(LIBRARY|PDS)         ]
      [UNIT(Unitname)               ]
      [VOLUME(volser)               ]
      [SUBSTITUTE(oldstring|newstring)]
      [RECFM(F|FB|FA|FM|FBA|FBM|V|VB|VA|VM|VBA|VBM|U|UA|U)]
```

Aliases

CR, CRE, CREA, CREAT, CREATE

Defaults

BLKSIZE, DIR, LRECL, OPTCD, SPACE, BLK/TRK/CYL, RECFM, ROUND, EXPDT/RETPD, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE as specified for the current data set.

Required

None

Operands

data set	The new data set name. Use standard TSO dsname syntax.
BLKSIZE(Bval)	A DCB BLKSIZE value for the new data set. *Bval* defaults to its value for the current data set or can be entered as a number from 0 through 32767.
DIR(num)	The number of directory blocks for the output data set. If you do not enter DIR, the default is taken from the current data set.
NODIR	Allocates the new data set as sequential regardless of the current data set.
LRECL(Lval)	A DCB LRECL value for the new data set. *Lval* defaults to its value for the current data set or can be entered as a number from 1 through 32767.
OPTCD(Newo)	The desired OPTCD value for the data set. *Newo* defaults to its value for the current data set or can be entered as W, WC, WCZ, C, CZ, or Z.
SPACE(prim,sec)	The primary and secondary space allocation quantities for the new data set in block, track, or cylinder units. If you do not enter BLK, TRK, or CYL but you do enter SPACE, the space allocation units are assumed to be the same as the current allocation.
BLK	Allocates the output data set in blocks. If you enter BLK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.
TRK	Allocates the output data set in tracks. If you enter TRK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.
CYL	Allocates the output data set in cylinder units. If you enter CYL without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.
ROUND	Used with BLK to specify that the data set allocation be rounded up to cylinder boundaries.
EXPDT(yyddd)	The expiration date for the data set. On and after the expiration date, delete or write over the data set. Also use the form EXPDT(yyyyddd).
RETPD(nnnn)	The number of days (0 to 9999) that the data set be retained by your data center.
STORCLAS(sc)	The storage class for the data set. Use this parameter to identify performance and availability requirements for data sets. Use this parameter instead of UNIT and VOLUME keywords for SMS managed data sets to determine where the data set is allocated. To nullify this parameter, type **STORCLAS(*).**
MGMTCLAS(mc)	The management class for the data set. This parameter establishes the migration, backup and space release characteristics for SMS managed data sets. To nullify this parameter, type **MGMTCLAS(*).**

DATACLAS(dc)	The data class for the data set. Use this parameter to provide data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL, and some VSAM SHR options for SMS managed data sets. To nullify this parameter, type DATACLAS(*). Use DATACLAS for data sets that are not SMS managed.
DSNTYPE(typ)	*typ* is specified as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS managed data set.
UNIT(Uname)	The allocation unit name. Use this parameter if the default unit name is incorrect for this allocation.
RECFM(Newr)	A DCB RECFM value for the new data set. *Newr* defaults to its value for the current data set or can be entered as F, FB, FA, FBA, FM, FBM, V, VB, VA, VBA, VM, VBM, U, UA, or UM.
VOLUME(volser)	The output volume name. Use this parameter if the data set should be allocated on a particular volume. Use the form VOLUME(*) if you want to make another allocation on the same volume used for the current data set.
SUBSTITUTE(old,new)	Used with a forward slash (/) to represent the current data set name. Use SUBSTITUTE to replace a string of characters in the source data set name to create a target data set name. For example, if the current data set is PDSEXXX.MSGS, the following two subcommands are equivalent: **CREATE PDSEYYYY.MSGS** **CREATE / SUBSTITUTE(XXX/YYYY)**

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
---------------------------- ISPMODE Session Display ---------  ROW 440 OF 446
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=: ------------------------------
>----->c lib.test
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80   9040     3X       47      10   10 TRK     25

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST,VOL=SER=STR801  MEM=:

>----->create lib.test2 spa(10,5)
PDS174I 'C911407.LIB.TEST2' has been created on volume STR815
>----->c lib.test2
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80   9040     1X       10       9    5 TRK     29

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=:
***************************** BOTTOM OF DATA  *****************************

(Note: the parameters specified with the CREATE command will be determined by the
     DFSMS environment, if one exists.)
```

```
-------------------------- MODEL entry panel -----------------------------
OPTION  ===>

 New DSNAME   ===> 'WSER07.SMSLINK.LOAD'
    on Volume ===> STG002    (optional)
    or  Unit  ===>           (optional)
 Space units  ===> CYL       (BLK, TRK or CYL)
 Round space  ===> NO        (YES or NO to round up to nearest CYL)
 Primary space===> 44                CURRENT Allocated space:   660     TRKS
 Secondary    ===> 5                         Unused space:        0     TRKS
 Directory blk===> 4                         Used directory blk: 4
 RECFM        ===> U                         DSORG:         PE
 LRECL        ===> 80                        Allocation type:   CYL
 BLKSIZE      ===> 32760                     Secondary quantity: 5
 Expiration date=>          (Julian)         Number of extents: 1
 Retention time =>          (number of days, this overrides EXPDT)
 Data class   ===>                    _____
 MGMT class   ===> STRG              |Press RCHANGE to save in PROFILE
 Storage class ==> STANDARD          |Press END to exit with no changes
 DSNTYPE      ===> LIBRARY (PDS/LIBRARY) |Press ENTER with non-blank DSNAME
                                     |      to create a new data set
                                     |Enter B on the OPTION line to select
                                     |      BLKSIZE modeling service
```

CSECTS Function

The CSECTS command displays a map of a load module in an ISPF table.

Example

CSECTS   idcams   unres

Syntax

CSECTS   member   [UNRESOLVED]

Aliases

CS, CSE, CSEC, CSECT, CSECTS

Operands

member	The member whose CSECTs and ENTRYs are to be mapped.
UNRESOLVED	Adds unresolved and weak unresolved references to the map.

Remarks

The CSECTS ISPF table displays in response to a CSECTS or NUCMAP command. When you are in a CSECTS display, delete a part of the table, sort the table in different

directions, find data in the table, print a part of the table or store a part of the table in a data set, and so on. For help with the different options, use the HELP command, CUA functions, or enter an O command as a primary command or as a line command.

The CSECTS table displays the names of all CSECTs and ENTRYs for a load module. Enter line commands against CSECT table lines that apply only to that particular CSECT. The screen fields for each line are as follows:

CMD	Command. Type a line command here.
TYPE	Type of entry: - CSECT - CSECT entry - ENTRY - ENTRY within a CSECT - COMMON - COMMON area - PROCEED - private area. This is an unnamed CSECT. - UNRESV - unresolved symbol. This symbol was required but was missing when the module was last linked. - WEAK-U - weak unresolved symbol. This symbol was not required and was missing when the module was last linked
DATA/MSG	Data field. Type up to eight bytes of operand information here. It is used as a command feedback field — any information in this field that is preceded by an asterisk (*) is ignored when you enter a line command. This field is set to *ENTRY for the entry point of the module.
SECTION	CSECT name.
ENTRY	ENTRY name or unresolved reference name.
ADDR	Address of this symbol.
LENG	Hexadecimal length of this CSECT.
MD/SG	Mode or segment. A segment number is available for overlay modules. For recently linked modules, an AMODE/RMODE indication is available, as follows: **R64 - RMODE of ANY and AMODE 64** - A64 - RMODE of 24 and AMODE 64 **RANY** - RMODE of ANY and AMODE 31 **AANY** - RMODE of 24 and AMODE ANY **A31** - RMODE of 24 and AMODE 31 **A24** - RMODE of 24 and AMODE 24
MEM ADDR	Memory address (this is for NUCMAP).

The following primary commands are supported directly for the CSECTS function. For information on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
CS[ECTS]	Rebuilds the CSECTS table.	

	Syntax	`CSECTS member [UNRESOLVED]`
EDITT[BL] (or ET[BL])	Enters an edit session on CSECT table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	
	Syntax	`F anystring [ASIS]` `[FIRST|LAST|PREV]   [PREFIX|SUFFIX|WORD]` `[ANY|ADDRESS|NAME|ENTRY|SEGMENT|MODE]`
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax	`LOCATE csectname|address`
O[PTIONS]	Provides primary command selection for the CSECTS function and operand syntax assistance.	
OUT[PUT]	Outputs the CSECTS table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the CSECTS table based on a string match.	
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]` `[ANY|ADDRESS|NAME|ENTRY|SEGMENT|MODE]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [ADDRESS|NAME]` `[ASCEND|DESCEND]`
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. The CSECTS UT panel (the same panel as the log) allows dynamic primary commands with which you specify command names and their corresponding actions.	
X	Clears the CSECTS table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the CSECTS table; this is equivalent to X ALL	

The following line commands are supported in the CSECTS function:

+	Provides an extension panel for command entry. *cm ERR- t="Line command,+">
=	Repeats the previous line command. *cm ERR- Tallinn command,=">
BASE	Assigns a base value at this CSECT for a ZAP line command on a later CSECT.

DELI	DELINK the CSECT.
DIS	DISASM the CSECT.
F	FIND data in the CSECT.
H	HISTORY for the CSECT.
K	Kills and clears all following line commands.
L	LIST data from the CSECT.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
O	Provides line command selection and operand syntax assistance.
R	REPLACE data in the CSECT.
S	Selects line command (normally LIST, it is set by SETSEL).
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
UT	Selects the extended user line command panel.
X	Drops the table line.
ZAP	Invokes the ZAP function for this CSECT.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

==	= command. Repeats the previous line command for the range of lines. *cm ERR- t="Block line command,=="">
LL	LIST command. Lists data from each CSECT.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
XX	X command. Drops the range of table lines.

```
 FUNCTIONS   CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS   FEATURES
------------------------ CSECT Display for STARTOOL  ----------   ROW 1 OF 137
COMMAND ===>                                                  SCROLL ==> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=STARTOOL --------------------------
CMD  TYPE   DATA/MSG SECTION   ENTRY     -ADDR- -LENG- MD/SG  MEM ADDR
     WEAK-U                    VTSOCMD
     WEAK-U                    PDS#OPT4
     WEAK-U                    PDS#SECI
     CSECT           PDSMAIN             000000 002E4C A24
     ENTRY                     PDSKLEAR  000138        A24
     ENTRY                     PDSNCMD   0006D0        A24
     ENTRY                     PDSSCAN4  000B06        A24
     ENTRY                     PDSCONVD  0011DC        A24
     ENTRY                     PDSLDATA  00144A        A24
     ENTRY                     PDSOPENX  0020CC        A24
     ENTRY                     PDSRETUR  002320        A24
     CSECT           PDSPDSIN            002E50 000D9F A24
     CSECT           PDSATTNX            003BF0 00015E A24
     CSECT           PDSSTAEX            003D50 0002EA A24
     CSECT           PDSALLOC            004040 00087D A24
     CSECT           PDSEXCP             0048C0 000BDE A24
     CSECT           PDSMSGS             0054A0 003412 A24
     CSECT  *ENTRY   PDSMAINA            0088B8 0001A0 A24
     CSECT  *ENTRY   TESTAM64            000000 00004C RANY
```

Commands — D to E

DCF Subcommand

The DCF subcommand prints a hardcopy list of a member. Use the TSO SCRIPT command. Add any SCRIPT operands after the member name.

Example

```
DCF    mema:memb  twopass  index  file(myfile)
```

Syntax

```
DCF  memgroup
        BIND(obind {ebind})           PAGE{({PROMPT}{{FROM} p {TO} q}
        CHARS(font1..font4)           {{FROM} p FOR n}{{FROM} p ON})}
        CONTINUE|NOCONT               PRINT{(copies,class,fcb,ucs)}
        CTF                           PROFILE{(fileid)}|NOPROFILE
        DDUT|NODDUT                   QUIET
        DEST(station-id)              SEARCH(libname)(opnum...)
        DEVICE(devtype)               SEGLIB|NOSEGLIB
        FILE{(fileid)}                SPELLCHK
        FONTLIB({ftype}{fmodel})      STOP
        FPASSES n                     SYON|SYOFF
        INDEX                         SYSVAR(n value...)
        LIB(libename...)(opnum...)    TERM
        MESSAGE({DELAY}{D}{TRACE})     TLIB
        NOSPIE                        TWOPASS
        NOWAIT                        UNFORMAT
        NUMBER                        UPCASE
        OPTIONS{(fileid)}             @user-option
```

Aliases

```
DC, DCF
```

Defaults

memgroup

Required

Operands

memgroup	The members to be printed. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
others	Optional. May include any desired TSO SCRIPT operands.

Remarks

This interface is optional. Use it only if DCF Script is installed.

DDNAME Function

Use the DDNAME command to display a subset of your TSO session allocations in the LISTA table. Maintain the DDNAME/LISTA table in DDNAME and concatenation order (the SORT command is not supported).

Example

```
DDNAME   isp*
```

Syntax

```
DDNAME namemask
```

Aliases

```
DD, DDN, DDNA, DDNAM, DDNAME
```

Defaults

Required

namemask

Operands

namemask	This is a partial DDNAME that may contain masking characters (***** for combinations, **/** for patterns and **:** for ranges). For example, ISP*LIB, PROC/SYS, and ISP:ISR.

Remarks

The DDNAME/LISTA ISPF table displays in response to a DDNAME command. When you are in a DDNAME/LISTA display, delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, and so on. For help with the different options available, use the HELP command, CUA functions, or enter an O command as either a primary command or as a line command.

DDNAME has alternate panels, as do most of the ISPMODE tables in StarTool FDM, that contain extra information about the listed data sets. The first DDNAME/LISTA panel shows basic allocation information such as DDNAME, data set TYPE, Open count (under the header O#), VOLUME allocated, MEMBER name, and DSNAME. Command error messages are provided under the DATA/MSG header as follows:

*INVALID	Line command is not supported.
*DD GONE	File is not allocated.
*OPEN DD	File is open.
*IN USE	File is in use.
*NOT CON	File is not concatenated.
*PERMCON	File is permanently concatenated.
NO DDN	File has a blank ddname.

TYPE is the data set DSORG or VIO, CTLG, VVDS, JES, or TERM depending on the type of allocation.

The second DDNAME/LISTA panel is a double line panel that includes some additional information: DISP for data set disposition and STAT for data set status. The STAT field is TMP for temporary, PRM for permanent, or CNV for convertible.

The following primary commands are supported directly for the LISTA/DDNAME function. For information on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ALT[ERNAT]	Displays an alternate view of the LISTA/DDNAME table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
BAT[CHJCL]	Builds batch JCL that includes all DD statements in the DDNAME table.	
COMPR[ESS]	Global command. Changes to each data set in the table and issues a COMPRESS subcommand.	
	Syntax	`COMPRESS [LIST|NOLIST|SUMMARY]`
DUA[L]	Displays a double line view of the LISTA/DDNAME table.	
DD[NAME]	Rebuilds the LISTA/DDNAME table selecting data sets with a DDNAME mask.	
	Syntax	`DDNAME ddnamemask`
EDITT[BL] (or ET[BL])	Enters an edit session on LISTA/DDNAME table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	

F	Finds a string and positions the display start location.	

	Syntax	```F anystring [ASIS]```
		``` [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]```
		``` [ANY|DDNAME|DSNAME|VOLUME|MEMBER|ALLOC]```

FI[ND]	Global command. Changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is applied to each data set individually, do not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.	

	Syntax	```FIND memgroup 'anystring'```
		``` [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]```
		``` [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]```

GL[OBAL]	Global command. Changes to each data set in the table and issues a StarTool FDM subcommand.	
	Syntax	```GL anysubcommand [anyoperands]```
LE[FT]	Rotates through alternate views of the LISTA/DDNAME table. PF keys 10 and 22 are set to LEFT.	
LISTA	Rebuilds the LISTA/DDNAME table selecting data sets by attributes.	
	Syntax	```TA [attributes]```
MODEL	Global command. Changes to each data set in the table and issues a MODEL command.	
NOR[MAL]	Displays the default view of the LISTA/DDNAME table.	
OUT[PUT]	Outputs the LISTA/DDNAME table to print or to a data set.	
	Syntax	```OUTPUT [=c	F(ddname)]```
REM[OVE]	Trims the LISTA/DDNAME table based on a string match.	
	Syntax	```REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]``` ``` [ANY|DDNAME|DSNAME|VOLUME|MEMBER|ALLOC]```
REPL[ACE]	Global command. Changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is applied to each data set individually, do not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.	

	Syntax	```REPLACE memgroup 'fromstring' 'tostring'```
		``` [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]```
		``` [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]```
		``` [WRITE|NOWRITE]```

RESET	Used with data set tagging, clears the DATA/MSG field in all table entries.	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the LISTA/DDNAME table. PF keys 11 and 23 are normally set to RIGHT.	
SEEK	Global command. Changes to each data set in the table and looks for a member.	
	Syntax	```SEEK member```

TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as at your installation.	
X	Clears the LISTA/DDNAME table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the LISTA/DDNAME table; equivalent to X ALL	

The following line commands are supported in the LISTA/DDNAME function:

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
6	Changes to the data set and issues a MEMLIST all.
A	Changes to the data set and executes an alternate subcommand.
B	Changes to the data set and issues a MEMLIST all.
C	Changes to the data set.
COPY	Changes to the data set and invokes the COPY subcommand.
DCAT	Deconcatenates nonpermanent data sets in a concatenated group.
DUP	Changes to the data set and invokes the DUP subcommand.
E	Changes to the data set and issues a MEMLIST all.
FREE	Frees a DDNAME or all data sets in a concatenated group.
GO	Changes to the data set using GO processing (a number is optional).
IDC	Performs an IDCAMS LISTC and directs output to the log.
K	Kills and clears all following line commands.
LC	Adds the data set name to the current LISTC/LISTF table.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST all.
MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
S	Selects line command (normally CHANGE, it is set by SETSEL).
SEEK	Changes to the data set and checks for a member.
SETA	Saves a StarTool FDM subcommand for repeated use by the ALT line command.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
U	Changes to the data set and displays USAGE.

UT	Selects the extended user line command panel.
W	Adds the data set name (and any member name) to the WORKPAD.
WHO	Checks for users of this data set with the WHOHAS subcommand.
X	Drops the table line.
XCOP	Changes to the data set and invokes the XCOPY interface.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTA/DDNAME function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR- t="Block line command,==">`
AA	A command. Changes and executes an alternate subcommand for each line.
BB	B command. Changes and issues a MEMLIST all for each line.
CC	C command. Changes data set for each line.
EE	E command. Changes and issues a MEMLIST all for each line.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	M command. Changes and issues a MEMLIST all for each line.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
SSEE	SEEK command. Changes to each data set and checks for a member.
UU	U command. Changes and displays USAGE for each line.
XX	X command. Drops the range of table lines.
ZZ	Z command. Changes and issues a COMPRESS for each line.

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS FEATURE
-------------------------- List Allocations -----------------  ROW 1 TO 7 OF 18
COMMAND ===>                                                    SCROLL ==> CSR
- DSN=C911407.LIB.ASM,VOL=SER=SER815  MEM=DAFX*  ----------------------------
CMD  DDNAME    DATA/MSG I/O's O# VOLUME  MEMBER   ------ DATA SET NAME --------
     ISPPLIB            177 1  SER006           SER07.LIB.PDSE
       #    2             2 1  SER007           SER.COMPAREX.PANELS
       #    3             0 1  TSG303           SYSI.IBMPDF.SYS350.PREPLIB
       #    4            27 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
     ISPMLIB              4 1  SER006           SER07.LIB.PDSE
       #    2             0 1  SER007           SER.COMPAREX.MSGS
       #    3             3 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
       #    4            88 1  TSG309           SYSI.IBMPDF.SYS350.ISPENU
       #    5             0 1  TSG301           SYSP.STD.ISPMLIB
     ISPSLIB              0 1  TSG301           SYSP.STD.ISPSLIB
       #    2             0 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
ut   ISPTLIB              1 1  TSG301           SYSP.STD.ISPTLIB
       #    2             0 1  TSG309           SYSI.IBMPDF.SYS350.ISRLIB
     ISPPROF             22 1  SYSR2C           SER07.ISPF.PROFILE
     ISP05482             8 1  SER004           SER07.SPFLOG2.LIST
```

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
_____
EDIT       WSER07.SPFTEMP1.CNTL                     Columns 00001 00072
Command ===>                                         Scroll ===> CSR
****** **************************** Top of Data ****************************
000001 //WSER07A JOB (X170,374),'CLIST CONCATENATION',CLASS=A,
000002 //   MSGCLASS=X,TIME=(0,30),NOTIFY=WSER07
000003 //PDA05021 EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=5M
000004 //SYSPROC  DD DISP=SHR,DSN=SYS1.SISPCLIB
000005 //*          UNIT=SYSALLDA,VOL=SER=OS39R2
000006 //          DD DISP=SHR,DSN=SYS1.CLIST
000007 //*          UNIT=SYSALLDA,VOL=SER=SCPMV5
000008 //          DD DISP=SHR,DSN=WSER07.LIB.CLIST
000009 //*          UNIT=SYSALLDA,VOL=SER=SER002
000010 //          DD DISP=SHR,DSN=SYS1.DGTCLIB
000011 //*          UNIT=SYSALLDA,VOL=SER=OS39R2
000012 //          DD DISP=SHR,DSN=SYS1.HRFCLST
000013 //*          UNIT=SYSALLDA,VOL=SER=OS39R2
```

```
---------------------- ut for LA   user line commands ----------------------
OPTION  ===>

Choose one of the following:
    AMS  - IDCAMS LISTC for data set 'SYSP.STD.ISPTLIB'
    DSAT - DSAT command for data set 'SYSP.STD.ISPTLIB'


Dynamic Commands--note:  < = TSG301,
                         / = 'SYSP.STD.ISPTLIB'
LAST - TRP(DSAT / LAST)
LD__ - TRP(LISTD / HISTORY)  /* PROVIDE HISTORY DATA
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
____ -
____ -
```

DECODE Subcommand

The DECODE subcommand decrypts a member. The data encryption algorithm follows the Advanced Encryption Standard (AES) set forth in Federal Information Processing Standard (FIPS) 197.

Example

```
DECODE 'USER1.C227693.CODED.U' 'USER1.CLEAR.COPY' 'USER1.C227693.KEY' SHR
```

Syntax

```
DECODE input-dsn output-dsn key-dsn [SHR|OLD]
```

Aliases

DEC, DECO, DECOD, DECODE

Defaults

OLD

Required

input-dsn, output-dsn, key-dsn

Operands

input-dsn	Fully qualified data set name enclosed in single quotes.
output-dsn	Fully qualified data set name enclosed in single quotes.
key-dsn	Fully qualified data set name enclosed in single quotes.
OLD	This allocation is not to be shared with any other allocation for this data set.
SHR	This allocation is to be shared with other allocations for this data set.

Remarks

FDM supports both decryption and encryption of data sets with the National Institute of Standards and Technology (NIST) publication FIPS 197. Details are in publication:

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Key Specification

The FDM users are custodians of their own keys. The data set that contains the keys can be RACF protected "for your eyes only". FDM will not show the keys in printed output or on the joblog or system console. FDM will erase the keys in memory so that they cannot be observed in program dumps and abends.

The key file contains exactly 2 records:

- Record 1 is the key that is used to process the input file.

- Record 2 is not used.

The key file is required for both decrypt and encrypt operations and is the same format in both cases. The actual key is expected to be a hexadecimal string. If you prefer a

character form, enclose the key specification in quotes. The format of the key in record 1 is:

Item	Column Position	Description
Key Type	1-3	Values can be: 000, 128, 192, 256. 000 means no key.
Space	4	Value must be a single space.
Key	5-68	Hexadecimal key specified in hex characters 0-9, lowercase a-f, or uppercase A-F. Maximum key length is: ■ 128 key type means key contains a maximum of 32 hexadecimal digits (16 hexadecimal bytes). ■ 192 key type means key contains a maximum of 48 hexadecimal digits (24 hexadecimal bytes). ■ 256 key type means key contains a maximum of 64 hexadecimal digits (32 hexadecimal bytes). Enclose character-format key in single quotes. ■ 128 key type means the key contains up to 16 alphanumeric characters. ■ 192 key type means the key contains from 17 to 24 alphanumeric characters. ■ 256 key type means the key contains from 25 to 32 alphanumeric characters.

Example Key Specifications

■ For a character key:

```
128 'CIPHER KEY'
```

■ For a hexadecimal key:

```
192 aabbccddee00999ff0ff
```

Decode Operation

FDM will decrypt the input file. If the key type is 000, the results that are written to the output data set remain unaltered.

Standard Verification Patterns

There is, published within the NIST spec, a simple set of patterns that verify the accuracy of the deciphering logic. There is a sample job that repeats this exercise. Users should verify the installation prior to using the feature. Once encrypted, file contents can never be recovered unless the proper keys are presented.

DELETE Subcommand

The DELETE (or SCRATCH) subcommand deletes a member from the data set. If you delete a member group, the members are deleted in reverse member group order to optimize directory operations.

If you are planning to delete members in a LLA controlled library, see "LLA and Deleted Members" in Chapter 6, "Commands — L to N".

Example

```
DELETE    mema:memb
```

Syntax

```
DELETE    memgroup [ALIAS|NOALIAS] [FORCE]
```

Aliases

```
DEL, DELE, DELET, DELETE, SCRA, SCRAT, SCRATC, SCRATCH
```

Defaults

memgroup, NOALIAS

Required

Operands

memgroup	Identifies the members to be deleted.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
ALIAS	Deletes any associated members (aliases, apparent aliases, and associated main members).
NOALIAS	Deletes only the members specifically identified.
FORCE	Deletes even if an edit session is currently using the member.

Remarks

NOTE This subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection".

```
-------------------------- ISPMODE Session Display --------  ROW 2636 OF 2647
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=DAF/  ----------------------------
>----->del daf/ alias
PDS162I Members to be deleted are: DAFABC, DAFJCLO, DISDAFD

PDS163I Associated members to be deleted are: ALIAS1

PDS394A Should all of these members be deleted (Y/N) ?
>----->y
PDS040I ALIAS1 has been deleted
PDS040I DISDAFD has been deleted
PDS040I DAFJCLO has been deleted
```

DELINK Subcommand

The DELINK subcommand reconstructs object code from a load module.

Example

DELINK   mymemb

Syntax

DELINK   memgroup [MODULE({* | Fullm | Partm*)}]

Aliases

DELI, DELIN, DELINK

Defaults

memgroup

Required

Operands

memgroup	Identifies the members to be decoded to object code.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT or ENTRY names to be considered.
	The MODULE operand has several valid forms:
	■ MODULE(*) – uses the previous name entered on any MODULE keyword
	■ MODULE(Fullm) – decodes only a CSECT or ENTRY named FULLM
	■ MODULE(Partm*) – decodes only a CSECT or ENTRY named PARTM

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session Display ----------------  ROW 1 OF 354
COMMAND ===>                                                    SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=VTOC  ----------------------------
>----->delink vtoc
//VTOC      EXEC PGM=IEWL,
//    PARM='NCAL,MAP,LIST,LET,RENT,REUS,REFR'
//SYSUT1     DD UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSPRINT   DD SYSOUT=*
//SYSLMOD    DD DISP=SHR,DSN=SER07.LINK.LOAD
//SYSLIBX    DD *
 ESD          VTOCCMD        VTOCMSG        VTOCEXCP
 ESD          VTOCSORT      PCLMAIN
 TXT            00  VTOCCMD   ...
```

DIRENTRY Subcommand

The DIRENTRY subcommand displays a member's directory entry.

Example

```
DIRENTRY   mema:memb
```

Syntax

```
DIRENTRY   memgroup [SHORT|LONG]
```

Aliases

```
DIR, DIRE, DIREN, DIRENT, DIRENTR, DIRENTRY
```

Defaults

memgroup, SHORT for THEN(DIR) or ELSE(DIR); otherwise, LONG

Required

Operands

memgroup	Identifies the members whose directory entry is to be displayed.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
SHORT	Provides only a directory entry dump.
LONG	Provides a directory entry dump and a display of individual elements from the directory entry.

```
------------------------ ISPMODE Session Display -----------  ROW 2601 OF 2630
COMMAND ===>                                                        L ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=PDS99  -------------------------
>----->direntry
PDS143I PDS99 Directory entry, length=46
        0000  D7C4E2F9 F9404040  010907B1 01091700  *PDS99    ........*
        0010  00000000 C2E303A9  B844A800 0000A800  *....BT.z..y...y.*
        0020  02000000 D7C4E2F9  F9C14040 0100      *....PDS99A  ..*

PDS262I LOC NAME     VALUE      DESCRIPTION
PDS262I --- ----     -----      -----------
PDS262I 00  PDS2NAME PDS99      MEMBER NAME
PDS262I 08  PDS2TTRP 010907     TTR OF FIRST BLOCK OF DATA
PDS262I 0B  PDS2INDC B1         ALIAS; 1 TTRS FOLLOW; 17 HALFWORDS OF DATA
PDS262I 0C  PDS2TTRT 010917,00  TTR OF FIRST TEXT BLOCK
PDS262I 10  PDS2TTRN 000000,00  (NOT USED FOR THIS MEMBER)
PDS262I 14  PDS2ATR1 C2         REENTRANT; REUS; NOT OVERLAY; NOT TEST
PDS262I                         NOT ONLY LOAD; NOT SCATTER; EXEC; NOT 1 TEXT
PDS262I 15  PDS2ATR2 E3         NOT DC; TEXT ORG=0; EP=0; HAS RLDS
PDS262I                         EDIT; NOT TEST; LKED F; REFRESHABLE
PDS262I 1F  PDS2FTB2 11         RMODE ANY; ALIAS AMODE 24; MAIN AMODE 64
```

DISASM Subcommand

The DISASM subcommand reconstructs assembler language instructions from a module. This procedure is often referred to as decoding or disassembling compiled or assembled instructions.

Example

```
DISASM MYMEMB OFFSET(1AF2)
```

Syntax

```
DISASM    memgroup [MODULE(name)]
                   [OFFSET(hexnum)]
```

```
[FLOAT|NOFLOAT]
[64BIT|NO64BIT]
[PRIV|SEMIPRIV|NOPRIV]
[REASM]
```

Aliases

DISA, DISAS

Defaults

memgroup, NOFLOAT, NOPRIV, NO64BIT

Required

None

Operands

memgroup	Identifies the members to be disassembled.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
MODULE(*name***)**	Specifies a 1- to 8-byte partial external name that limits CSECT or ENTRY names to be displayed.
	If you enter both MODULE and OFFSET keywords, the offset applies to each module selected. The MODULE operand has several valid forms:
	■ MODULE(*) – uses the previous name entered on any MODULE keyword.
	■ MODULE(PAR) – formats only a CSECT or ENTRY named PAR.
	■ MODULE(PAR*) – formats only a CSECT or ENTRY who name begins with PAR.
OFFSET(*hexnum***)**	Specifies a 1- to 6-digit hexadecimal module offset at which the display is to begin. If you enter both MODULE and OFFSET keywords, the offset applies to each module selected.
FLOAT	Translate floating point instructions in the instruction set.
NOFLOAT	Do not translate floating point instructions in the instruction set.
64BIT	Translate instructions that deal with 64-bit registers and addresses.
NO64BIT	Translate instructions that only deal with 32-bit registers and addresses.
PRIV	Translate privileged and semiprivileged instructions in the instruction set.
SEMIPRIV	Translate only semiprivileged instructions in the instruction set. Semi-privileged instructions are those generally not allowed in certain environments such as CICS and IMS. Examples are: PC and PR.

NOPRIV	Translate neither privileged nor semiprivileged instructions in the instruction set.
REASM	Outputs the disassembled instructions in a format that can be submitted to recreate the decoded module. When you request this format, the module offset field and the CSECT offset field (the first two fields of output) are reversed in order and placed at the end of each decoded statement. PDS141I header messages are suppressed and JCL is created to assemble and link the decoded module and its aliases.

Remarks

The DISASM subcommand disassembles load members and formats its data in one of two ways:

■ Non-REASM format:

```
000 0000 FBILDV00 CSECT
000 0000 FBILDV00 AMODE 31
000 0000 FBILDV00 RMODE 24
000 0000          B     32(,R15)              47F0 F020        *.00.*
004 0004          DC    X'1A',C'F'            1AC6             *.F*
```

■ REASM format (usable for reassembling):

```
//DIV       EXEC REASMLNK,OUTPUT='??REASM.OUTPUT',
//  PARM.LKED='NCAL,MAP,LIST,LET,RENT,OVLY,TEST'
//ASM.SYSIN   DD *
FBILDV00 CSECT                                              000 0000
FBILDV00 AMODE 31                                           000 0000
FBILDV00 RMODE 24                                           000 0000
         B     32(,R15)              47F0 F020     *.00.*    000 0000
         DC    X'1A',C'F'            1AC6          *.F*      004 0004
         DC    X'0000000000000000'   0000000000000000  *........* 006 0006
```

The individual data columns are as follows:

module offset	A hexadecimal offset in the current module. With REASM format, this field is the last output field.
CSECT offset	A hexadecimal offset in the current CSECT. With REASM format, this field is just before the last output field.
name	An external name or generated internal name (the letter A followed by a 6-character offset in the module).
operation	1- to 7-byte reconstructed operation code.
operands	The reconstructed operand values.
comments	For SVC calls, the original MACRO (for example, GETMAIN or TPUT).
hex representation	1 to 3 columns of the original hexadecimal data.
character representation	1 to 8 bytes of the same data in character mode surrounded by asterisks.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session Display ----------  ROW 8936 OF 10720
COMMAND ===>                                                 ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAPHELP  -----------------------
>----->disasm zaphelp off(40c0)
PDS141I AT 003FF0  CSECT CSOUT     LENGTH 000428
0040C0 00D0           BAL   R15,228(,R12)          45F0 C0E4        *.0U*
0040C4 00D4           DC    AL4(A0040CC)           000040CC         *.. .*
0040C8 00D8           DC    X'00000000'            00000000         *....*
0040CC 00DC  A0040CC  DC    C'IKJDAIR '            C9D2D1C4C1C9D940 *IKJDAIR*
0040D4 00E4           SVC   6      LINK            0A06             *..*
0040D6 00E6           BXH   R15,R15,58(R12)        86FF C03A        *f..*
0040DA 00EA           MVC   144(8,R10),416(R13)    D207 A090 D1A0   *K...J.*
0040E0 00F0           XC    16(4,R10),16(R10)      D703 A010 A010   *P.....*
0040E6 00F6           LA    R0,48(,R10)            4100 A030        *....*
0040EA 00FA           ST    R0,8(,R10)             5000 A008        *&...*
0040EE 00FE           MVI   8(R10),X'8F'           928F A008        *k...*
0040F2 0102           MVC   48(96,R10),936(R12)    D25F A030 C3A8   *Kª..Cy*
0040F8 0108           MVC   88(8,R10),144(R10)     D207 A058 A090   *K.....*
0040FE 010E           LA    R1,8(,R10)             4110 A008        *....*
004102 0112           SVC   19     OPEN            0A13             *..*
```

DISPLAY Subcommand

The DISPLAY subcommand lists member names from the data set directory. The default member group is not affected by the DISPLAY or PATTERN subcommands.

Example
DISPLAY   aa   b4

Syntax
DISPLAY   name1 name2

Aliases
D, DI, DIS, DISP, DISPL, DISPLA, DISPLAY

Defaults

entire directory

Required

Operands

name1	The starting member name (or portion of the member name) at which the display is to start. If omitted, the entire directory displays. Enter a group name descriptor such as part1:part2, seg1/seg2, or part1*seg1 for name1, in which case, the output consists of the member names that are displayed by the MEMBERS subcommand.
name2	Specifies the ending member name (or portion of the member name) after which the display is to stop. If omitted, the display continues to the end of the directory.

Remarks

Neither DISPLAY nor PATTERN affect the current member group. If you enter DISPLAY without operands, all members in the data set display. This differs from PATTERN in that PATTERN remembers its previously entered operands.

-A is added following a displayed member name if the member name is an alias. The member name is listed in a combined hexadecimal/character format if it does not contain upper-case alphanumeric characters (including @, $, and #) or if the first character is numeric. Unprintable characters (for 3270-type devices) in a member name display as periods.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session Display ----------------  ROW 518 OF 520
COMMAND ===>                                                           OLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=(PDSR --------------------------
>----->d @*
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK
>----->d @:d
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK     ALIAS1  -A  ALLGLOB     ASMEXT2     CHECKOUT
CICS        COMPARE     COMPARES    COMPA1      COMPA2      COMPA3
DISASM3     DISDAFD     DISDAFE
```

DSAT Subcommand

This interface is optional, and is used only if your installation installed the public domain DSAT command. For current information on the DSAT command, refer to the CBT Web site at http://www.cbttape.org/, file 112.

Use the DSAT subcommand to display allocation information for data sets on a direct access device.

DSAT searches the catalog for the entries for the data sets specified. Allocation information is obtained from the volume table of contents, formatted and displayed. If a name is an index name, all data sets below the index display.

Bypass the catalog search by supplying the volume name on which the data set resides. This option permits displaying information for uncataloged data sets.

Example

```
DSAT lib
```

Syntax

```
DSAT [(dsnamelist)]   [ALL|DAONLY                          ]
                      [ALLOC|NOALLOC                        ]
                      [CCHHR|NOCCHHR                        ]
                      [CRDATE|NOCRDATE                      ]
                      [DCB|NODCB                            ]
                      [DEVICE|NODEVICE                      ]
                      [DEVTYPE|NODEVTYPE                    ]
                      [DSORG|NODSORG                        ]
                      [EXDATE|NOEXDATE                      ]
                      [GDGDATA|NOGDGDATA                    ]
                      [GENERIC|DSONLY                       ]
                      [HARDCOPY(ddname)                     ]
                      [HEADER|NOHEADER                      ]
                      [LASTREF|NOLASTREF                    ]
                      [PDS|NOPDS                            ]
                      [PRINT|NOPRINT                        ]
                      [RC(TALLOC|TUSED|TDIFF|               ]
                          LALLOC|LUSED|LDIFF|NUM|           ]
                          PREVIOUS|DSORG|DIRALLOC|          ]
                          DIRUSED|ENTRIES|ALIASES|MEMBERS) ]
                      [SECONDARY|NOSECONDARY                ]
                      [SEQNO|NOSEQNO                        ]
                      [SERIAL|NOSERIAL                      ]
                      [TOTALS|NOTOTALS                      ]
                      [VOLUME(name)                         ]
```

Aliases

```
DSA, DSAT
```

Defaults

dsnamelist defaults to 'prefix' to list a user's data sets; see underscores above.

If you specify HARDCOPY, NOPRINT is the default.

Required

Operands

dsnamelist	One or more data set or index level names. Use TSO naming conventions. If a specified name is an index name, data sets under that level display. If you do not specify name, the prefix is used. If no name is specified and the profile specifies NOPREFIX, the user ID is used.
	If you enter a keyword, you must specify *dsnamelist*. If you omit *dsnamelist*, it does not default to the prefix, but the keyword is interpreted as the *dsnamelist*, not a keyword.
ALL	Displays all data set names under an index level.
DAONLY	Displays data sets on mounted direct access devices.
ALLOC	Lists allocation information (tracks allocated, tracks free, and extents).
NOALLOC	Does not list allocation information.
CCHHR	Displays Format 1 DSCB address.
NOCCHHR	Does not display CCHHR.
CRDATE	Lists creation date.
NOCRDATE	Does not list creation date.
DCB	Lists DCB information (RECFM, BLKSIZE, and LRECL).
NODCB	Does not list DCB information.
DEVTYPE	Displays device type code (for example, 3010200F for a 3390).
NODEVTYPE	Does not display device type code.
DEVICE	Lists device name (3380, 9345, etc.).
NODEVICE	Does not list device name.
DSORG	Lists data set organization.
NODSORG	Does not list data set organization.
EXDATE	Lists expiration date.
NOEXDATE	Does not list expiration date.
GDGDATA	Displays information on generation data group indexes.
NOGDGDATA	Does not display information on generation data group indexes.
GENERIC	Name specifies a generic key.
DSONLY	Treats names as data set names, not index names.
HARDCOPY(dd)	Specifies the DDNAME of a pre-allocated data set for output from DSAT.
HEADER	Displays an output header line.
NOHEADER	Does not display a header line.
LASTREF	Displays last date referenced.
NOLASTREF	Does not display last date referenced.
PDS	Lists PDS directory information (blocks alloc, used, entries, and aliases).

NOPDS	Does not list PDS directory information.
PRINT	Output is to be displayed.
NOPRINT	Output is not to be displayed, except for error messages.
RC(type)	How the return code is to be set.

Continuing RC(type):

If you use DSAT as a StarTool FDM subcommand, return code processing is not performed (the return code is set to zero). If you use DSAT in a CLIST or REXX exec or in single subcommand mode as in the following example, the return code is set properly:

```
STARTOOL FILE(ISPPROF) DSAT LIB
RC(TALLOC)
```

If you do not specify RC, RC=4 means successful completion and RC=16 means unable to interpret command. Otherwise, RC is set to a value as specified by:

- **TALLOC -** total tracks allocated
- **TUSED -** total tracks used
- **TDIFF -** total over-allocation (TALLOC minus TUSED)
- **LALLOC -** tracks allocated by last data set
- **LUSED -** tracks used by last data set
- **LDIFF -** last data set over-allocation (LALLOC minus LUSED)
- **NUM -** number of data sets displayed (including data sets not found)
- **PREVIOUS -** assumes the return code of the previous command
- **DSORG -** sets the return code based on the data set organization of the data sets displayed as:
- All data sets were sequential
- All data sets were PDS
- Data sets were mixed PDS and sequential
- Data set other than PDS or sequential or an error
- **DIRALLOC -** number of directory blocks allocated
- **DIRUSED -** number of directory blocks used

	- **ENTRIES -** number of entries in the directory (members and aliases)
	- **ALIASES -** number of alias entries
	- **MEMBERS -** number of entries less the number of aliases
SECONDARY	Displays secondary allocation information.
NOSECONDARY	Does not display secondary allocation information.
SEQNO	Lists file sequence numbers.
NOSEQNO	Does not list file sequence numbers.
SERIAL	Lists volume names.
NOSERIAL	Does not list volume names.
TOTALS	Displays totals (tracks allocated, tracks free, and data sets displayed).

NOTOTALS	Does not display totals.
VOLUME	Specifies the volume to be searched for the data set.

Remarks

Use DSAT in CLISTs to find the allocation of a data set or a group of data sets and set the return code to the specified value. Test the return code with the WHEN command. Suppress output by specifying NOPRINT. Use the HARDCOPY keyword to direct the output to a pre-allocated data set.

Select the attributes to be displayed by specifying keyword operands. The information displayed is:

- Volume name on which the data set is located.

- File sequence number

- Device type code from catalog entry

- Allocation (allocated, free, and extents)

- Secondary allocation (amount and units)

- Data set organization

- DCB (RECFM, BLKSIZE, and LRECL)

- Creation date

- Expiration date

- Last date referenced

- Fully qualified data set name

- CCHHR of the Format 1 DSCB

- Generation data group data

- PDS directory information

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------- ISPMODE Session# 1 Log# 1 ------------  ROW 61 TO 77 OF 77
COMMAND ===>                                                  L ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  ------------------------
>----->dsat lib
SERIAL ALLOC  FREE EX DSORG -DCB ATTRIBUTES- CR. DATE -DSNAME-
SER006   379     0  2 A-PO  FB   13680    80 03/05/93 SER07.LIB.ASM
SER007   140    97  4 A-PO  FB    9040    80 10/04/89 SER07.LIB.CLIST
SER006     4     2  3 A-PO  VB   32760   255 03/15/91 SER07.LIB.CLISTV
SER006   100    15  1 A-PO  FB   13680    80 03/17/93 SER07.LIB.CNTL
SER006     1     0  1 A-PO  FB    9040    80 07/17/92 SER07.LIB.EXPDT
SER007    25     0  1 A-PO  FB    9040    80 10/04/89 SER07.LIB.FILE112
SER007    63     0  1 A-PO  FB    9040    80 10/04/89 SER07.LIB.FILE296
SER007   203     0  2 A-PO  U    32000     0 10/04/89 SER07.LIB.LOAD
SER006   765   225  3 A-PO  FB   13680    80 05/06/93 SER07.LIB.PDSE
SER006   459     0  1 A-PO  FB   13680    80 05/06/93 SER07.LIB.PDSE311
SER006   180     0  2 A-PO  FB   13680    80 06/29/93 SER07.LIB.PDSE320
SER007     4     0  1 A-PO  FB    9040    80 09/21/92 SER07.LIB.PDSINSTL
SER007     2     0  1 A-PO  U    13030     0 07/09/93 SER07.LIB.STOW
SER007    10     9  1 A-PS  FB    2000   200 06/30/93 SER07.LIB.TESTVSAM
        2335   348                          14 DATA SETS DISPLAYED
****************************** BOTTOM OF DATA ******************************
```

DSNAME Subcommand

The DSNAME subcommand displays statistics on the current data set allocation using
message format PDS200I, PDS210I, or PDS220I depending on the last DSNAME operand
(MSG, TSO, or JCL) entered.

The DSNAME subcommand automatically invokes while executing several subcommands
such as in CHANGE, COMPRESS, FIXPDS, and USAGE. MSG format output always displays
VSAM data sets.

Example

DSN

Syntax

DSNAME   [MSG | TSO | JCL]

Aliases

DS, DSN, DSNA, DSNAM, DSNAME, DSNAMES

Defaults

Initially MSG; thereafter, previous MSG/TSO/JCL

Required

Operands

MSG	Uses the PDS200I message (table or msg format) for allocation display.
TABLE	Uses the PDS200I message (table or msg format) for allocation display.
TSO	Uses the PDS210I message (TSO format) for allocation display.
JCL	Uses the PDS220I message (JCL or batch format) for allocation display.
BATCH	Uses the PDS220I message (JCL or batch format) for allocation display.

```
----------------------- ISPMODE Session Display ------------- ROW 2648 OF 2661
COMMAND ===>                                                    OLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=DAF/  ----------------------------
>----->dsname jcl
PDS220I //SYS00134  DD  DSN=C911407.LIB.TEST,DISP=SHR,UNIT=3380,
PDS220I //  DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040,OPTCD=C),VOL=SER=STR815,
PDS220I //  SPACE=(TRK,(47,40,30))              /*FREE TRK=10,FREE DIR=25*/

>----->dsn tbl
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80   9040    1X    47      10     40 TRK      25

>----->dsn tso
PDS210I ALLOC F(SYS00134) DA('C911407.LIB.TEST') SHR UNIT(3380) -
PDS210I   RECFM(F B) LRECL(80) BLKSIZE(9040) OPTCD(C) VOLUME(STR815) -
PDS210I   TRK SPACE(47,40) DIR(30)              /*FREE TRK=10,FREE DIR=25*/
```

DUP Subcommand

The DUP subcommand copies members to another data set. It is in the COPY subcommand and performs the copy with internal routines. Use the COPY subcommand without concern for using DUP since DUP routines are used as needed.

Copying members between unlike data sets is supported; however, load members can only be copied to load libraries. Records are truncated or padded with blanks during the copy as required.

If the source data set is RECFM=F with LRECL=80 and the target data set is RECFM=V with LRECL=255 (or the reverse), DUP supports CLIST conversion during the copy. CLIST conversion involves assigning new sequence numbers at the beginning or end of each output line. For converting from RECFM=V to RECFM=F, DUP breaks lines up as required to fit in 80-character lines using a **-** as a continuation character.

If the data set to be copied contains load modules (RECFM is U) and the output block size is smaller than the input block size, DUP does not copy the data set since a COPYMOD function is required. In this case, use the COPY subcommand. COPY automatically requests a COPYMOD function because of the differences in block sizes.

Example

```
DUP   mem1:mem2  other.dataset   shr
```

Syntax

```
DUP   memgroup
      {/ or dsname [VOLUME(volser)] | * | FILE(ddname) [NUMBER(numb)]}
      [OLD|SHR | NEW [BLK|TRK|CYL]     [DIR(num)]   [SPACE sec)]]
               [ROUND]   [EXPDT(yyddd)|RETPD(nnnn)]   [RLSE]
               [STORCLAS(sclass)]   [MGMTCLAS(mclass)]
               [DATACLAS(dclass)]   [DSNTYPE(LIBRARY|PDS)]
               [UNCTLG|NOCATLG]
         [ALIAS|NOALIAS                    ]
         [CHANGE|GO                        ]
         [ADDZAP|ADDZAP(numb)              ]
         [NONUM|SNUM                       ]
         [NOCLIST                          ]
         [REXX                             ]
         [AS(pname) | TO(pname)            ]
         [CALL|NOCALL                      ]
         [EXIST|NOEXIST|EXCLUDE|NOBLDL     ]
         [MAXBLK(num)                      ]
         [REPLACE|NOREPLACE                ]
         [SUMMARY|LIST|NOLIST              ]
         [UNIT(Unitname)                   ]
         [CHANGE|GO                        ]
         [MOVE                             ]
         [SUBSTITUTE(oldstring,newstring)]
         [VOLUME(volser)                   ]
       * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)   ]
       * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)         ]
*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases

```
DU, DUP
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

ALIAS if load, NOALIAS otherwise; SHR, NOREPLACE, MAXBLK(output BLKSIZE). If NEW is specified, BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE default to values used for the input data set.

Required

memgroup, dsname | FILE(ddname)

Operands

memgroup	The members to be copied to the target data set. Default member names, member lists, member name ranges, and member name patterns are allowed. See Appendix A, "Formatting Member Names".
dsname	The output data set. Use standard TSO dsname syntax or a temporary data set name such as &&TEMP. Only sequential, direct or partitioned output data sets are supported by DUP. As an alternative to the AS or TO keywords, enter partial member names in parentheses following the data set name, such as: `the.dsn(pname)`
FILE(ddname)	The output data set is allocated with the provided DDNAME. Only disk data sets are supported. They must be sequential or partitioned.
NUMBER(numb)	The concatenation number for the DDNAME identified. *numb* defaults to 1. If *numb* is larger than the number of concatenated data sets, the last data set in the concatenation sequence is used.
OLD	Obtains exclusive use of the output data set.
SHR	Allows shared use of the output data set. StarTool FDM provides output data set protection. See Appendix B, "Update Protection".
CHANGE	Specifies that StarTool FDM should CHANGE to the output data set at the end of the DUP process.
GO	Specifies that StarTool FDM should initiate a GO session for the output data set at the end of the DUP process.
ALIAS	Copies any associated members (aliases, apparent aliases, and associated main members).
NOALIAS	Copies only the members specifically identified.
AS(pname)	*pname* defines the name of any output members. For example, with AS(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are created. AS and TO are equivalent keywords. If only a single member is input and NOALIAS is in effect, the resulting member name is taken only from the AS or TO keyword.
TO(pname)	*pname* defines the name of any output members. For example, with TO(new) and the input members AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are created. TO and AS are equivalent keywords. If only a single member is input and NOALIAS is in effect, the resulting member name is taken only from the AS or TO keyword.
REPLACE	Replaces members in the output data set by any copied members.
NOREPLACE	Does not replace members in the output data set by copied members.

ADDZAP	For load members only, specifies that a ZAP IDR record is to be added (each ZAP IDR record contains space for 19 historical entries).
ADDZAP(numb)	For load members only, add 1 to 99 ZAP IDR records.
MAXBLK(nm)	For source members only, *nm* specifies the maximum output block size to be used when the members are reconstructed.
NONUM	For source members only, specifies that the input members are all data and no sequence numbers are to be formatted. This keyword is ignored on a RECFM=F to RECFM=V CLIST conversion.
SNUM	For source members only, specifies that sequence numbers in the input (the first 8 columns for RECFM=V data sets; otherwise the last 8 columns) are to be discarded during the copy.
NOCLIST	CLIST conversion is not to be performed. Perform CLIST conversion if RECFM=V, LRECL=255 data is copied to RECFM=F, LRECL=80 or the reverse. CLIST conversion entails assigning new sequence numbers and breaking up long lines into several short ones while maintaining CLIST format rules.
REXX	Specifies that RECFM=VB, LRECL=255 REXX members be converted to RECFM=F, LRECL=80 REXX members using REXX continuation rules. This algorithm works for many REXX routines but it is not error-free because this subcommand is not aware of text strings delimited by "or".
UNIT(uname)	The allocation unit name. Use this parameter if the default unit name is incorrect for this allocation.
NEW	Creates the output data set. When you specify NEW, also specify BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT (or RETPD), RLSE, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE keywords.
VOLUME(volser)	For NEW data sets, specifies the output volume name. For existing data sets, Use this parameter if the output data set is not cataloged. For new data sets, code VOLUME(*) to mean the volume parameter is to be copied.
DIR(num)	The number of directory blocks for the output data set. If you do not enter DIR, the default is taken from the input data set.
SPACE(prim,sec)	The primary and secondary quantities for the output data set in track or cylinder units. If you do not enter BLK, TRK, or CYL but you enter SPACE, the space allocation units are assumed to be the same as the current allocation.
BLK	Allocates the output data set in blocks. If you enter BLK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.

TRK	Allocates the output data set in tracks. If you enter TRK without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.
CYL	Allocates the output data set in cylinder units. If you enter CYL without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.
ROUND	Used with BLK to specify that the data set allocation be rounded up to cylinder boundaries.
EXPDT(yyddd)	The expiration date for the data set. On and after the expiration date, delete or write over the data set. Also use the format EXPDT(yyyyddd).
RETPD(nnnn)	The number of days (0 to 9999) that the data set be retained by your data center.
RLSE	Any unused space in the data set is to be released after the copy operation.
STORCLAS(sc)	The storage class for the data set. Use this parameter to identify performance and availability requirements for data sets. Use this parameter instead of UNIT and VOLUME keywords for SMS-managed data sets to determine where the data set is allocated. To nullify this parameter, enter **STORCLAS(*).**
MGMTCLAS(mc)	The management class for the data set. This parameter establishes the migration, backup, and space release characteristics for SMS-managed data sets. To nullify this parameter, enter **MGMTCLAS(*).**
DATACLAS(dc)	The data class for the data set. Use this parameter to provide data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL, and some VSAM SHR options for SMS-managed data sets. To nullify this parameter, enter **DATACLAS(*)**. Use DATACLAS for data sets that are not SMS-managed.
DSNTYPE(typ)	**Specify *typ*** as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS-managed data set.
UNCATLG	Does not catalog the NEW data set.
NOCATLG	Does not catalog the NEW data set.
CALL	Calls the copy program to perform the copy.
NOCALL	Does not call the copy program. Member checking, control card generation, and data set allocations are performed.
EXIST	Copies only the subset of memgroup that is also present in the output data set (the REPLACE keyword is also used).
NOEXIST	Copies only the subset of memgroup that is *not* present in the output data set.
NOBLDL	Does not perform BLDL (member existence) checking in the target library. Uses this parameter when performance of the COPY subcommand is very important and BLDL checking (by StarTool FDM) for possible errors is not important.
SUMMARY	Displays only copy summary and error messages.

LIST	Displays all copy messages.
NOLIST	Displays no copy messages.
MOVE	If no errors are detected during the copy process, all of the copied members are deleted.

SUBSTITUTE(old,new)	Used with a **/** to represent the current data set name. Use SUBSTITUTE to replace a string of characters in the source data set name to create a target data set name.
	
	 For example, if the current data set is PDSEXXX.MSGS, the following two subcommands are equivalent:
	 `DUP ABC* PDSEYYYY.MSGS`
	 `DUP ABC* / SUBSTITUTE(XXX/YYYY)`

FROMKEY(ky)	For VSAM data sets only, code ky as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. Specify it only for an alternate index or a key-sequenced data set.
FROMADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, code ky as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.

TOADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.

Remarks

If you use the TO (or the equivalent AS) keyword, each member is renamed in the copy in a generic fashion. For example, if you enter TO(X12), each resulting member name is formed by placing X12 over the first three characters of the original member name. An exception to this rule is made when only a single member is to be copied and its aliases are not also requested. In this case, the member name is taken directly from the TO keyword.

For load members, add a null ZAP IDR record as the member is being copied. For non-load members, the output member is reblocked to the receiving data set's BLKSIZE or the MAXBLK specified.

NOTE This subcommand modifies the output data set. To ensure data set integrity, allocate the output data set as OLD, or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection" on page 425.

```
-------------------- ISPMODE Session# 1 Log# 1 ---------- ROW 153 TO 161 OF 161
 COMMAND ===>                                                         OLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.CLISTV,VOL=SER=STR92T  MEM=AA* ---------------------------
 >----->dup aa* lib.clistv2
 PDS072I CLIST conversion will be performed
 PDS852E AA already exists
 PDS051I AAAA was copied; input=6; output=6
 PDS051I AAC was copied; input=44; output=44
 PDS051I AA2 was copied; input=1; output=1
 PDS051I AA6 was copied; input=70; output=70

 PDS054I Totals -- Members=4; Input=121; Characters=9,680
```

```
-------------------- ISPMODE Session# 1 Log# 1 ------------- Row 75 to 93 of 93
COMMAND ===>                                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=STARXYZ* -------------------------
>------>dup starxyz* lib.cntls replace
PDS051I STARXYZE was replaced; input=27; output=27
PDS051I STARXYZJ was copied; input=15; output=15
PDS051I STARXYZM was copied; input=26; output=26
PDS051I STARXYZO was copied; input=25; output=25
PDS051I STARXYZT was copied; input=22; output=22
PDS051I STARXYZU was copied; input=12; output=12
PDS051I STARXYZV was copied; input=39; output=39
PDS051I STARXYZ1 was replaced; input=14; output=14
PDS051I STARXYZ2 was replaced; input=22; output=22
PDS051I STARXYZ3 was copied; input=12; output=12
PDS051I STARXYZ5 was copied; input=19; output=19
PDS051I STARXYZ6 was copied; input=21; output=21
PDS051I STARXYZ8 was copied; input=14; output=14
PDS051I STARXYZ9 was copied; input=23; output=23

PDS054I Totals -- Members=14; Input=291; Characters=23,280
```

DVOL Subcommand

This interface is optional. Use it only if your installation installed the public domain DVOL command. For information on the DVOL command, refer to the CBT Web site at http://www.cbttape.org/, file 112.

Use the DVOL subcommand to display the amount of free space on a direct access device.

Example

DVOL tso long

Syntax

```
DVOL [(volumelist)|* ] [SHORT|LONG       ]
                       [HEADER|NOHEADER ]
```

```
[PRINT|NOPRINT   ]
[SERONLY         ]
[HARDCOPY(ddname) ]
```

Aliases

DV, DVO, DVOL

Defaults

Use an ***** to display all volumes, SHORT, HEADER, and PRINT. If HARDCOPY is specified, NOPRINT is the default.

Required

Operands

volumelist	A list of one or more volume names of mounted direct access volumes or partial volume names.
	If you do not enter a volume, or if you specify an *****, the information displays for each direct access volume on the system.
	If you enter a volume name with a **U** followed by 1 to 3 digits, the data is used as a UCB name.
SHORT	Displays short form output.
LONG	Displays long form output.
HEADER	Displays a short form header.
NOHEADER	Displays no short form header.
PRINT	Displays output.
NOPRINT	Displays no output.
SERONLY	Displays only a list of mounted volumes.
HARDCOPY(dd)	Specifies the DDNAME of a pre-allocated data set to receive a copy of DVOL output.

Remarks

DVOL reads the Format 4 and Format 5 DISCOs from the VTOC of a direct access volume and displays:

- Volume name

- Unit address

- Device type including density indicators

- Mount status and use status

- Number of blank DISCOs in the VTOC

- Condition of the VTOC indicators byte

- VSAM data fields

- Total free space in tracks

- Number of free extents

- Number of free cylinders

- Size of largest extents (up to 5) in cylinders + tracks

- Size of largest extents (up to 5) in tracks

The Return code is set to the total number of tracks in the 5 largest extents (to a maximum of 4095) for the last volume displayed.

If an error condition exists on the volume, the return code is set to 0.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
---------------------- ISPMODE Session# 1 Log# 1 ------------  ROW 46 TO 60 OF 60
COMMAND ===>                                                    OLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  ------------------------
>----->dvol abc
SERIAL  --UNIT--  ATTRIBUTES  VSAM AVAIL -----TOTALS---- LARGEST-EXTENT 5 EXTS
        ADR TYPE   MOUNT/USE        DSCBS TRACKS   EXT CYL CYL+TR  TRACKS TRACKS
ABC522 522 3380E  RESDNT/PRIV OFF   791  13247     3 881  442+14   6644  13247
ABCD4B 531 3390M3 RESDNT/PRIV OFF   753   9538    14 632  422+ 0   6330   9399

>----->dvol abcd4b long
VOLUME SERIAL = ABCD4B     UNIT = 531     TYPE = 3390M3
STATUS: RESIDENT   PRIVATE  UNALLOCATED  ONLINE
VTOC CONTAINS  753 BLANK DSCB'S
INDEXED VTOC CONTAINS   86 FREE VIR'S
FREE SPACE IS  9538 TRACKS IN   14 EXTENTS INCLUDING  632 FULL CYLINDER
LARGEST EXTENTS ARE: /CYL+TR/ 422+ 0/ 100+11/  60+ 3/  40+ 0/   3+10/
                     /TRACKS/   6330/   1511/    903/    600/     55/
****************************** BOTTOM OF DATA ******************************
```

EDIT Subcommand

The EDIT or SPFEDIT subcommand enters ISPF edit for a member.

Example
```
EDIT    mema:memb
```

Syntax
```
EDIT    memgroup [OLD|NEW                        ]
                 [MACRO(macname)                 ]
                 [SM(macname)|SETMACRO(macname)  ]
                 [PROFILE(profname)              ]
                 [SP(profname)|SETPROF(profname) ]
                 [CONFIRM|NOCONFIRM              ]
                 [SETCONFIRM|NOSETCONFIRM        ]
               * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)   ]
```

```
                     * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)          ]
```

*NOTE: Lines with an asterisk are supported for VSAM with PEDIT or EDIF.

Aliases

E, ED, EDI, EDIT, S, SP, SPF, SPFE, SPFED, SPFEDI, SPFEDIT

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

memgroup, confirm

Required

Operands

memgroup	The source members to be edited.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
OLD	Verifies that the member exists before invoking ISPF EDIT service.
NEW	Verifies that the member does not exist before invoking ISPF EDIT service.
MACRO(mac)	An initial edit macro name.
SM(mac)	A default initial edit macro name. To nullify this parameter, enter SM(*).
SETMACRO(mac)	A default initial edit macro name. To nullify this parameter, enter SETMACRO(*).
PROFILE(prof)	The edit profile name (the supplied name is used instead of the low-level DSNAME qualifier).
SP(prof)	A default edit profile name. To nullify this parameter, enter SP(*).
SETPROF(prof)	A default edit profile name. To nullify this parameter, enter SETPROF(*).
CONFIRM	For ISPF 4.x, specifies that you desire prompting for MOVE, REPLACE, or CANCEL operations.
NOCONFIRM	For ISPF 4.x., no confirmation prompting for MOVE, REPLACE, or CANCEL operations.

SETCONFIRM	For ISPF 4.x, specifies a default of CONFIRM for subsequent edits in this StarTool FDM session. Override this keyword on individual edit commands with NOCONFIRM.
NOSETCONFIRM	For ISPF 4.x, specifies a default of NOCONFIRM for subsequent edits in this StarTool FDM session. Override this keyword on individual edit sessions with CONFIRM.
FROMKEY(ky)	For VSAM data sets only, code ky as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. Specify it only for an alternate index or a key-sequenced data set.
FROMADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, code ky as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.

TOADDRESS(ad)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.

Remarks

On the first entry of an EDIT or SPFEDIT subcommand, StarTool FDM presents any pending EDIT recovery sessions to you before entering EDIT for a selected member.

If you enter a **:** for the member name position, a MEMLIST is provided instead.

VSAM data sets are edited with the PEDIT command. However, use ISPF EDIF services to access up to 255 characters of any individual record, but the SAVE command is disabled. You may have chosen a different interface during StarTool FDM installation. Of the interfaces supported, only PEDIT and EDIF support positioning keywords such as FROMKEY.

To determine which interface is used, type a CONTROL DEFAULT subcommand and look for an output line beginning "EDIT calls" with one of the following values:

EDIF - uses the ISPF EDIF interface with up to 255 characters per record

PEDIT - uses PEDIT services

%VSAMMED - uses the MacKinney System'VSAM Utility Edit

EDREC Subcommand

The EDREC subcommand explicitly invokes ISPF edit recovery.

Example
EDREC

Syntax

```
EDREC
```

Aliases

```
EDR, EDRE, EDREC
```

Operands

No operands are supported for the EDREC subcommand.

Remarks

This subcommand allows you to control when EDIT RECOVERY is to take place. EDIT RECOVERY procedures are initiated on the first EDIT subcommand. With the EDREC subcommand you can perform the appropriate action before the first EDIT subcommand.

ENCODE Subcommand

The ENCODE subcommand encrypts a member. The data encryption algorithm follows the Advanced Encryption Standard (AES) set forth in Federal Information Processing Standard (FIPS) 197.

Example

```
ENCODE 'USER1.C227693.CLEAR.VB' 'USER1.C227693.CODED.U' 'USER1.C227693.KEY' SHR
```

Syntax

```
ENCODE input-dsn output-dsn key-dsn [SHR|OLD]
```

Aliases

```
ENC, ENCO, ENCOD, ENCODE
```

Defaults

OLD

Required

input-dsn, output-dsn, key-dsn

Operands

input-dsn	Fully qualified data set name enclosed in single quotes.
output-dsn	Fully qualified data set name enclosed in single quotes.
key-dsn	Fully qualified data set name enclosed in single quotes.
OLD	This allocation is not to be shared with any other allocation for this data set.
SHR	This allocation is to be shared with other allocations for this data set.

Remarks

FDM supports both decryption and encryption of data sets with the National Institute of Standards and Technology (NIST) publication FIPS 197. Details are in publication:

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

When you compare encrypted files without decrypting, that is at the binary level, a single byte difference within a record will cause the entire record to be different. This is a feature of the encryption algorithm.

Key Specification

The FDM users are custodians of their own keys. The data set that contains the keys can be RACF protected "for your eyes only". FDM will not show the keys in printed output or on the joblog or system console. FDM will erase the keys in memory so that they cannot be observed in program dumps and abends.

The key file contains exactly 2 records:

- Record 1 is the key that is used to process the input file.

- Record 2 is not used.

The key file is required for both decrypt and encrypt operations and is the same format in both cases. The actual key is expected to be a hexadecimal string. If you prefer a character form, enclose the key specification in quotes. The format of the key in record 1 is:

Item	Column Position	Description
Key Type	1-3	Values can be: 000, 128, 192, 256. 000 means no key.

Item	Column Position	Description
Space	4	Value must be a single space.
Key	5-68	Hexadecimal key specified in hex characters 0-9, lowercase a-f, or uppercase A-F. Maximum key length is: ■ 128 key type means key contains a maximum of 32 hexadecimal digits (16 hexadecimal bytes). ■ 192 key type means key contains a maximum of 48 hexadecimal digits (24 hexadecimal bytes). ■ 256 key type means key contains a maximum of 64 hexadecimal digits (32 hexadecimal bytes). Enclose character-format key in single quotes. ■ 128 key type means the key contains up to 16 alphanumeric characters. ■ 192 key type means the key contains from 17 to 24 alphanumeric characters. ■ 256 key type means the key contains from 25 to 32 alphanumeric characters.

Example Key Specifications

■ For a character key:

```
128 'CIPHER KEY'
```

■ For a hexadecimal key:

```
192 aabbccddee00999ff0ff
```

Encode Operation

FDM will encrypt the input file. If the key type is 000, the results that are written to the output data set remain unaltered.

Standard Verification Patterns

There is, published within the NIST spec, a simple set of patterns that verify the accuracy of the deciphering logic. There is a sample job that repeats this exercise. Users should verify the installation prior to using the feature. Once encrypted, file contents can never be recovered unless the proper keys are presented.

END Subcommand

The END subcommand terminates the STARTOOL command. In ISPMODE, the END subcommand terminates the current function. Multiple END subcommands are required to terminate the STARTOOL command.

Example

```
END
```

Syntax

```
END
```

Aliases

```
EN, END
```

Operands

No operands are supported for the END subcommand.

Remarks

In ISPMODE or MEMLIST an END terminates the current function instead of the STARTOOL command. To terminate StarTool FDM enter repeated END commands or a QUIT subcommand.

EXCLUDE Subcommand

The EXCLUDE subcommand removes member names from the current MEMLIST. Use EXCLUDE by itself or as follows:

```
FIND isp* 'string' else(exclude)

   or

IF    mem| alias .... then(exclude)
```

Example

```
EXCLUDE  isp*  norent
```

Syntax

```
EXCLUDE memgroup
        [SINCE|BEFORE                          ]
        [TODAY|YESTERDAY|WEEK|CURRENT|BIWEEK|
           MONTH|QUARTER|HALFYEAR|YEAR|BIYEAR|
           LAST(numdays)|DATE(yyyy/mm/dd)      ]
        [CHANGED(yyyy/mm/dd:yyyy/mm/dd)         ]
        [CREATED(yyyy/mm/dd:yyyy/mm/dd)         ]
        [ABOVE(Count1)                          ]
        [ALIAS|NOALIAS                          ]
        [AMODE24|AMODE31|AMODE64|AMODEANY|NOAMODE24|
           NOAMODE31|NOAMODE64|NOAMODEANY   ] (load only)
```

```
            [AUTH|NOAUTH                          ] (load only)
            [BELOW(Count2)                        ]
            [DC|NODC                              ] (load only)
            [EDIT|NOEDIT                          ] (load only)
            [EXEC|NOEXEC                          ] (load only)
            [FLEVEL|NOFLEVEL                      ] (load only)
            [HASALIAS|NOHASALIAS                  ]
            [ID(Puid)|NOID|NOTID(Puid)            ]
            [LKED(Partl)                          ] (load only)
            [LOADONLY|NOLOADONLY                  ] (load only)
            [MODULE({* | Fullm | Partm*)}         ] (load only)
            [NULL|NONULL                          ]
            [ORPHAN|NOORPHAN                      ]
            [OVERLAY|NOOVERLAY                    ] (load only)
            [PAGE|NOPAGE                          ] (load only)
            [REFR|NOREFR                          ] (load only)
        [RENT|NORENT                          ] (load only)
            [REUS|NOREUS                          ] (load only)
            [RMODE24|RMODEANY|NORMODE24|NORMODEANY] (load only)
            [SCTR|NOSCTR                          ] (load only)
            [SSI(hxdata)|SSI|NOSSI|PARTSSI(hxdata)]
            [SYSMOD(Partu) | USERDATA(Partu)      ] (load only)
            [TEST|NOTEST                          ] (load only)
            [TRANS(Partt)                         ] (load only)
            [TTR(Lttr:Httr)                       ]
            [USERDATA(Partu) | SYSMOD(Partu)      ] (load only)
            [USERID(Puid)|NOUSERID|NOTUSERID(Puid)]
            [VSLKED|NOVSLKED                      ] (load only)
            [ZAP(Partz)                           ] (load only)
```

Aliases

EXC, EXCL, EXCLU, EXCLUD, EXCLUDE

Defaults

memgroup, SINCE

Required

Operands

memgroup	The members to be excluded from the current MEMLIST.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names"
SINCE	Uses dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date. For load modules, it is the linkage edit date.

FROM	Uses dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date. For load modules, it is the linkage edit date.
BEFORE	Uses dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date. For load modules, it is the linkage edit date.
TO	Uses dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date. For load modules, it is the linkage edit date.
TODAY	A date, equivalent to LAST(0).
YESTERDAY	A date, equivalent to LAST(1).
WEEK	A date, equivalent to LAST(7).
CURRENT	A date, equivalent to LAST(10).
BIWEEK	A date, equivalent to LAST(14).
MONTH	A date, equivalent to LAST(30).
QUARTER	A date, equivalent to LAST(120).
HALFYEAR	A date, equivalent to LAST(183).
YEAR	A date, equivalent to LAST(365).
BIYEAR	A date, equivalent to LAST(730).
LAST(numdays)	A date, the number of days before today.
DATE(cdate)	The actual date. Enter it in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).
CHANGED(date:	Excludes members modified by an ISPF editor or a ZAP program between the dates specified. If you enter a single date, only that date is checked. A date range is entered like: `CHANGED(yyyy/mm/dd:yyyy/mm/dd)` Only members with ISPF statistics or members created by a linkage editor are supported.
CREATED(date:	Excludes members created with ISPF statistics or by the linkage editor between the dates specified. If you enter a single date, only that date is checked. A date range is entered like: `CREATED(yyyy/mm/dd:yyyy/mm/dd)`
ABOVE(Count1)	Excludes members with more than the number of lines (amount of storage for load modules) specified by the number, Count1.
ALIAS	Excludes alias members.
NOALIAS	Excludes main members.
AMODE24	Excludes modules with addressing mode 24.
NOAMODE24	Excludes modules with addressing mode 31, 64 or ANY.

AMODE31	Excludes modules with addressing mode 31.
NOAMODE31	Excludes modules with addressing mode 24, 64 or ANY.
AMODE64	Excludes modules with addressing mode 64.
NOAMODE64	Excludes modules with addressing mode 24, 31 or ANY.
AMODEANY	Excludes modules with addressing mode ANY.
NOAMODEANY	Excludes modules with addressing mode 24, 31 or 64.
AUTH	Excludes APF authorized modules.
NOAUTH	Excludes non-authorized modules.
BELOW(Count2)	Excludes members with less than the number of lines (amount of storage for load modules) specified by the number, Count2.
DC	Excludes downward-compatible modules.
NODC	Excludes non-downward-compatible modules.
EDIT	Excludes modules that are reprocessed by the linkage editor.
NOEDIT	Excludes modules that cannot be reprocessed by the linkage editor.
EXEC	Excludes EXECUTABLE modules.
NOEXEC	Excludes non-EXECUTABLE modules.
FLEVEL	Excludes modules processed by the F-level linkage editor.
NOFLEVEL	Excludes modules not processed by the F-level linkage editor.
HASALIAS	Excludes main members that have one or more aliases.
NOHASALIAS	Excludes alias members or main members without any aliases.
ID(Puid)	Excludes members with ISPF statistics and user IDs matching the partial name, Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. ID is an alias of USERID.
NOID	Excludes members without ISPF statistics. NOID is an alias of NOUSERID.
NOTID(Puid)	Excludes members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name, Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. NOTID is an alias of NOTUSERID.
LKED(Partl)	Excludes modules identified as linked by the linkage editor defined by the partial name, Partl. Enter Partl as a partial linkage editor IDR name with one to ten characters.
LOADONLY	Excludes modules marked for LOAD ONLY.
NOLOADONLY	Excludes modules not marked for LOAD ONLY.

MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT names for attribute searches. If you enter this parameter with any TRANS, SYSMOD/USERDATA, or ZAP keywords, only those CSECTs that satisfy the MODULE keyword are checked for the other keyword conditions. The MODULE operand has several valid forms:
	■ MODULE(*) – uses the previous name entered on any MODULE keyword
	■ MODULE(Fullm) – excludes any module containing a CSECT or ENTRY called FULLM
	■ MODULE(Partm*) – excludes any module containing a CSECT or ENTRY called PARTM...
ORPHAN	Excludes alias members that have no associated main member.
NOORPHAN	Excludes main members or alias members that have an associated main member.
OVERLAY	Excludes OVERLAY modules.
NOOVERLAY	Excludes non-OVERLAY modules.
PAGE	Excludes modules marked for loading on a page boundary.
NOPAGE	Excludes modules not marked for loading on a page boundary.
REFR	Excludes refreshable modules.
NOREFR	Excludes non-refreshable modules.
RENT	Excludes reentrant modules.
NORENT	Excludes non-reentrant modules.
REUS	Excludes reusable modules.
NOREUS	Excludes non-reusable modules.
RMODE24	Excludes modules with residence mode 24.
NORMODE24	Excludes modules with residence mode ANY.
RMODEANY	Excludes modules with residence mode ANY.
NORMODEANY	Excludes modules with residence mode 24.
SCTR	Excludes scatter-loaded modules.
NOSCTR	Excludes non-scatter-loaded modules.
SSI(hxdata)	Excludes members with matching SSI data.This is implemented as a generic search matching SSI characters from left to right for the number of characters entered.
SSI	Excludes members with SSI data.
NOSSI	Excludes members without SSI data.
PARTSSI(hxdata)	Excludes members with matching SSI data. This is implemented as a pattern search matching SSI digits as a string anywhere in the SSI field of the member.

SYSMOD(Partu)	Excludes modules with user IDR data that matches the partial name, Partu. Enter Partu as partial IDR data with one to eight characters. If you enter MODULE(...), the SYSMOD operand applies only to CSECTs whose names satisfy the MODULE argument.
TEST	Excludes modules with the TEST attribute.
NOTEST	Excludes modules without the TEST attribute.
TRANS(Partt)	Excludes modules with CSECTS identified as having been assembled or compiled by the translator defined by the partial name, Partt. Enter Partt as a partial translator IDR name with one to ten characters. If you enter MODULE(...), the TRANS operand applies only to CSECTs whose names satisfy the MODULE argument.
TTR(Lttr:Httr)	Excludes members whose start address is in the specified TTR range. *Lttr* defaults to 0 and can be entered as a hexadecimal TTR value from 0 through FFFFFF. *Httr* defaults to FFFFFF and can be entered as a hexadecimal TTR value from 0 through FFFFFF.
USERDATA(Partu)	Excludes modules with user IDR data that matches the partial name, Partu. Enter Partu as partial IDR data with one to eight characters. If you enter MODULE(...), the USERDATA operand applies only to CSECTs whose names satisfy MODULE.
USERID(Puid)	Excludes members with ISPF statistics and user IDs matching the partial name, Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. USERID is an alias of ID.
NOUSERID	Excludes members without ISPF statistics. NOUSERID is an alias of NOID.
NOTUSERID(Puid)	Excludes members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name, Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. NOTUSERID is an alias of NOTID.
VSLKED	Excludes modules linked by the system linkage editor.
NOVSLKED	Excludes modules not linked by the system linkage editor.
ZAP(Partz)	Excludes modules with zap IDR data that matches the partial name, Partz. Enter Partz as partial zap IDR data with one to eight characters. If you enter MODULE(...), the ZAP operand applies only to CSECTs whose names satisfy the MODULE argument.

Remarks

If a member to be excluded is not in the MEMLIST, no action is taken. The following sample screens show a before image of a member list and an after image showing the results of an EXCLUDE = RENT.

```
 FUNCTIONS   CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
---------------------- Load MEMLIST 1, Session# 1 ------- ROW 1 TO 16 OF 138
COMMAND ===> exclude = rent                                 SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=A/ --------------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED -- ATTRIBUTES -- APF  MODE  MAIN
     ADC2              *ALIAS   87/04/30 REFR RENT                   ADC2TSO
     ADC2ACF2                   88/06/02 REFR RENT
     ADC2RACF                   88/06/06 REFR RENT
     ADC2SUB                    87/03/23 REFR RENT
     ADC2SVC           *ALIAS   85/01/17 REFR RENT                   IGC0024I
     ADC2TDSM                   87/03/27 REFR RENT
     ADC2TSO                    87/04/30 REFR RENT
     AEV00                      89/07/11
     AEV10                      89/07/11 RENT REUS
     AEV20                      89/07/11 RENT REUS
     AEV50                      89/07/11
     AEV80                      89/07/11 RENT REUS
     AEV90                      89/07/11 RENT REUS
     ASID                       87/06/15                       RANY
```

```
 FUNCTIONS   CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
----------------------------------------------------------------------------
------------------------ Load MEMLIST 1, Session# 1 ----- ROW 1 TO 16 OF 91
COMMAND ===>                                                SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=(ADC2 ----------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED -- ATTRIBUTES  -- APF  MODE  MAIN
     AEV00                      89/07/11
     AEV50                      89/07/11
     ASID                       87/06/15                       RANY
     ASIDSP13                   82/05/17
     ASIDZZ           *ALIAS    82/05/17                              ASIDMM
     ASM                        78/02/19      DC
     ASMH                       82/05/25
     ASMH2                      82/05/25
     ASMTOZAP                   81/09/02
     AUTO                       82/06/16                  AC=1
     BLAKJACK                   82/05/18
     CACHE                      85/08/04                  AC=1
     CALCVSAM                   84/07/26
     CALENDAR                   89/03/06
```

EXEC Subcommand

The EXEC or % subcommand executes a CLIST containing StarTool FDM subcommands. Use the TSO CLIST processor. Both implicit (%...) and explicit (EXEC ...) call forms are supported.

Example
EXEC lib(clistmem) 'parms' list

Syntax

{EXEC   clistname ['operands'] | %clistmem [operands]}

Aliases

%, EX, EXE, EXEC

Defaults

Required

clistname | clistmem

Operands

clistname	The CLIST data set name.
clistmem	The name of a member in the CLIST partitioned data set allocated to FILE(SYSPROC).
operands	Optional. May include any CLIST parameters.

Remarks

The % (implied CLIST) and EXEC subcommands are supported in an ISPF dialog environment. Use the ISPF TSO command to invoke a CLIST that operates under ISPF (independent of StarTool FDM).

When you enter subcommands from a CLIST (or in batch mode or from a storage stack), no YES/NO prompts are provided; instead, YES responses are assumed in each case. YES responses are also assumed if CONTROL NOPROMPT is in effect.

Chapter 5
Commands — F to I

FILTER Function

Use the FILTER command to set selection values for use later by the LISTC/LISTF or MASK function. The NOFILTER keyword in MASK resets all FILTER options to their defaults.

Example
```
FILTER  novs
```

Syntax
```
FILTER   [ALL|P|PS|PO|PE|PDS|DA|IS|NOVS|VS|GDG]
         [ASSOC|NOASSOC                       ]
         [CHECKCAT|NOCHECK                     ]
         [MIGRAT|NOMIGRAT|ONLYMIGR             ]
         [OPTICAL|NOOPTICAL                    ]
         [QUICK|NOQUICK                        ]
         [RESET                                ]
```

Aliases
```
FIL, FILT,\ FILTE, FILTER
```

Defaults

ALL, ASSOC, NOCHECK, MIGRAT, NOOPTICAL, NOQUICK

Required

Operands

ALL	Default. Searches for all data set types.
P	Searches for PS, PO, and PE data sets.
PS	Searches for sequential data sets.
PO	Searches for partitioned data sets.
PE	Searches for partitioned extended data sets.
PDS	Searches for PO and PE data sets.
DA	Searches for direct data sets.
IS	Searches for ISAM data sets.
NOVS	Searches for non-VSAM data sets.
VS	Searches for VSAM data sets.
GDG	Searches for Generation data sets.
ASSOC	Default for LISTC. Adds associated components for VSAM clusters.
NOASSOC	For LISTC, does not add associated VSAM components.
CHECKCAT	For LISTF, checks the catalog status of each data set as it is added.
NOCHECK	Default for LISTF. Does not check catalog status.
MIGRAT	Default for LISTC. Searches for all data sets.
NOMIGRAT	For LISTC, searches for non-migrated data sets.
ONLYMIGR	For LISTC, searches for only migrated (MIGRAT or ARCIVE) data sets.
OPTICAL	Opens the VTOC for Optical devices (3395 M151).
NOOPTICAL	Default. Does not open the VTOC for optical devices (3395 M151).
QUICK	For LISTC, bypasses data set volume verification.
NOQUICK	Default for LISTC. Verifies data set volume status.
RESET	Resets the FILTER options to default values.

Remarks

Some special considerations apply for the FILTER command:

1   Migrated data sets are identified by a MIGRAT or ARCIVE volume name.

2   DSORG type checking cannot be performed on migrated data sets.

3   If QUICK is active, only GDG, NOVSAM and VSAM can be checked. P/PS/PO/PE/PDS/ DA/IS all result in non-VSAM data set selection.

4   MIGRAT/NOMIGRAT/ONLYMIGR, QUICK and ASSOC are only applicable to LISTC.

5   CHECKCAT/NOCHECK is applicable to LISTF only.

FIND Subcommand

The FIND subcommand displays portions of a member that contain a search string. Optionally, enter the THEN or ELSE keywords to allow conditional execution of other subcommands. When you enter a THEN keyword, the results of the FIND subcommand are not displayed unless you also enter the DISPLAY keyword.

Example

```
FIND membera:memberz    'this data'
```

Syntax

```
FIND    memgroup 'search_string'
        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]
        [CAPS|ASIS|IGNORE|PICTURE]
        [WORD|PREFIX|SUFFIX]
        [OFFSET(hexnum) MODULE({*|fullname|part*})]
        [AFTER(num) DO(num) MAXMEMBERS(num) COLS(from:to)]
        [AND('string2') ACOLS(from:to) {ACAPS|AASIS|AIGNORE|APICTURE}]
        [ORx('string3') OCOLS(from:to) {OCAPS|OASIS|OIGNORE|OPICTURE}]
        [FORMAT(from:to,from:to, ...)]
        [SKIPREC(num) MAXIN(num) MAXOUT(num) MAXFIND(num)]
        [SKIPCOL(num) MAXLEN(num) {FIRST|NOFIRST}]
        [DISPLAY]
        [THEN({ATTRIB|BROWSE|DIRENTRY|DELETE|EDIT|EXCLUDE|FSE|
               HISTORY|LIST|MAP|MEMBERS|MEMLIST|NEWML|PRINT|REVIEW|
               SUBMIT|SUBLIST|TSOEDIT|TSOLIST|VERIFY|VIEW})]
        [ELSE({ATTRIB|BROWSE|DIRENTRY|DELETE|EDIT|EXCLUDE|FSE|
               HISTORY|LIST|MAP|MEMBERS|MEMLIST|NEWML|PRINT|REVIEW|
               SUBMIT|SUBLIST|TSOEDIT|TSOLIST|VERIFY|VIEW})]

        [FROMKEY(key)|FROMADDRESS(address)|FROMNUMBER(num)][1]
        [TOKEY(key)|TOADDRESS(address)|TONUMBER(num)][1]
```

[1]NOTE: Supported for VSAM only.

Aliases

```
FI, FIN
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

memgroup, previous search string, IGNORE, NUM or previous LIST|FIND|REPLACE format

Required

None

Operands

memgroup	The members to be searched. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
'search_string'	The data to be located, delimited by single quotes. If you do not enter the string or if it is null, the last string entered is used. The string can contain 1 to 32 characters. If you enter the ASIS keyword, the string is not translated to upper-case letters. As an alternative, enter hexadecimal data as a string delimited with x's like x0123456789abcdefx. A hexadecimal string can contain 1 to 64 characters. *x333x* and *x0333x* are equivalent.
NUM	Examines the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record for numerics. If the line number field is not numeric, switch formatting to NONUM mode for the remainder of the member. For ISPF-saved members, the high-order 6 digits of the line number field is formatted. Otherwise, the low-order 6 digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from the logical record.
SNUM	Discards the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise). Searches or displays up to 256 bytes from a logical record.
NONUM	Searches or displays up to 256 bytes from a logical record (without regard to line numbers).
LBLOCK	Formats logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks. Only formatted segments that contain the search string display. However, a following segment also displays if the string spans a segment boundary. For load modules, only CSECT data is searched or displayed and if a name is entered in a MODULE operand, only those CSECT or ENTRY names that satisfy the name are searched or displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a **+** is placed after the key); otherwise, it is a relative byte address.

LDUMP	Formats logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. Only formatted segments that contain the search string display; however, a following segment also displays if the string spans a segment boundary. For load modules, only CSECT data is searched or displayed and if a name is entered in a MODULE operand, only those CSECT or ENTRY names that satisfy the name are searched or displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than eight characters, a **+** is placed after the key); otherwise, it is a relative byte address.
BLOCK	Formats physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks. Only formatted segments that contain the search string display; however, a following segment also displays if the string spans a segment boundary. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than eight characters, a **+** is placed after the key); otherwise, it is a relative byte address. If you perform control interval processing, the first field is always the control interval relative byte address. **TIP:** For FIND operations against very large data sets (several MB), use the BLOCK parameter to prevent ABENDs.
DUMP	Formats physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. Only formatted segments that contain the search string display; however, a following segment displays if the string spans a segment boundary. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a **+** is placed after the key); otherwise, it is a relative byte address. If you perform control interval processing, the first field is always the control interval relative byte address.
CAPS	Upper case any character string search argument.
ASIS	Does not upper-case the search string.
IGNORE	Searches for upper- and lower-case string arguments.

PICTURE	If specified, search for data using matching characters and the following picture characters:
	■ = for any character
	■ % for any alphanumeric character
	■ @ for any alphabetic character
	■ # for any numeric character
	■ $ for any special character
	■ ~ for any non-blank character (can also use the "not" character)
	■ . for any invalid character
	■ - for any non-numeric character
	■ < for any lower-case alphabetic character
	■ > for any upper-case alphabetic character
WORD	Searches for strings preceded and followed by a non-alphanumeric character. Can be used to limit the string hits.
PREFIX	Searches for strings preceded by a non-alphanumeric character. Can be used to limit the string hits.
SUFFIX	Searches for strings followed by a non-alphanumeric character. Can be used to limit the string hits.
OFFSET(*hexnum***)**	A 1- to 6-digit hexadecimal module offset at which the search is to begin. This operand is defined only for load modules and only with the LDUMP or LBLOCK output formats. If you enter MODULE and OFFSET keywords, the offset applies to each module selected.
MODULE(*name***)**	A 1- to 8-byte partial external name that limits CSECT or ENTRY names to be searched or displayed. This operand is defined only for load modules and only with the LDUMP or LBLOCK output formats. If you enter MODULE and OFFSET keywords, the offset applies to each module selected. The MODULE operand has several valid forms: ■ **MODULE(*)** – uses the previous name entered on any MODULE keyword ■ **MODULE(Fullm)** – searches or display only a CSECT or ENTRY named FULLM ■ **MODULE(Partm*)** – searches or display only a CSECT or ENTRY named PARTM.
AFTER(*num***)**	Coded as 0 through 9999999 to specify the number of matches to skip before reporting FIND strings.
MAXMEMBERS(*num***)**	*n* is coded as 0 through 9999999 to specify the maximum number of members to be selected by FIND.
DO(*num***)**	Coded as 0 through 9999999 to specify the maximum number of lines to output for a member.

COLS(*from***:***to***)**	Column range to search for the primary string (COLS is only supported for formats NUM, SNUM, and NONUM). For example, COLS(1:10) specifies that the string may begin in columns 1 through 10.
AND('*string2***')**	Another string to search in addition to the primary string. AND strings are supported only for formats NUM, SNUM, or NONUM and a FIND is reported for a primary string only if the AND string is found first.
ACOLS(*from***:***to***)**	Column range to search for the AND string. For example, ACOLS(1:10) specifies that the string can begin in columns 1 through 10.
ACAPS	If you use an AND character string argument, it is translated to upper-case letters.
AASIS	If you use an AND character string argument, it is not to be translated to upper-case letters.
AIGNORE	If you use an AND character string argument, searches for upper- and lower-case data matching the string.
APICTURE	If you use an AND character string argument, search for data with matching characters and the following picture characters: ■ = for any character ■ % for any alphanumeric character ■ @ for any alphabetic character ■ # for any numeric character ■ $ for any special character ■ ~ for any non-blank character (can also use the "not" character) ■ . for any invalid character ■ - for any non-numeric character ■ < for any lower-case alphabetic character ■ > for any upper-case alphabetic character
ORx('*string3***')**	Another string to search as an alternate to the primary string. ORx can be OR and OR2 through OR7, for a total of eight OR conditions. OR strings are supported only for formats NUM, SNUM or NONUM. A FIND is reported for either a primary string or an OR string.
OCOLS(*from***:***to***)**	Column range to search for the OR string. For example, OCOLS(1:10) specifies that the string can begin in columns 1 through 10.
OCAPS	If you use an OR character string argument, it is translated to upper-case letters.
OASIS	If you use an OR character string argument, it is not to be translated to upper-case letters.
OIGNORE	If you use an OR character string argument, search for upper- and lower-case data matching the string.

OPICTURE	If you use an OR character string argument, searches for data with matching characters and the following picture characters:
	■ = for any character
	■ % for any alphanumeric character
	■ @ for any alphabetic character
	■ # for any numeric character
	■ $ for any special character
	■ ~ for any non-blank character (can also use the "not" character)
	■ . for any invalid character
	■ - for any non-numeric character
	■ < for any lower-case alphabetic character
	■ > for any upper-case alphabetic character
FORMAT (*from*:*to*,*from*:*to*, **...**)	Specifies the record columns to display in the output when reporting a FIND string. FORMAT is supported only with NUM, SNUM, or NONUM data. For example, FORMAT(21:30,0,1:10) specifies that the output be formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.
SKIPREC(*num***)**	Ignores *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.
MAXIN(*num***)**	Inputs up to *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.
MAXOUT(*num***)**	Displays up to *n* (coded as 0 through 9999999) output lines for a member.
MAXFIND(*num***)**	Locates up to *n* (coded as 0 through 9999999) strings in a member.
SKIPCOL(*num***)**	Ignores *n* (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats).
	For NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.
MAXLEN(*num***)**	Searches or displays up to *n* (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).
FIRST	Lists all remaining data in the member after locating a string.
NOFIRST	Lists only string matches.
DISPLAY	The results of the FIND subcommand display before taking any THEN action. This allows you to see the lines containing the string and take a conditional action.

THEN(*action***)**	Executes a subcommand if the string is found in a member. The subcommand can be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY, or VIEW. NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET.
ELSE(*action***)**	Executes a subcommand if the string is *not* found in a member. The subcommand may be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY, or VIEW. NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET.
FROMKEY(*key***)**	For VSAM data sets only, code ky is coded as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. Specify it only for an alternate index or a key-sequenced data set.
FROMADDRESS (*address***)**	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components. ■ Can be specified for any VSAM data set component if control interval processing is being used. ■ Cannot be specified if the data set is being accessed through a path. ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.
FROMNUMBER(*num***)**	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it only for a variable or fixed relative record data set.
TOKEY(*key***)**	For VSAM data sets only, code ky as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.

TOADDRESS (*address*)	For VSAM data sets only, code ad as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record (or a control interval for control interval processing). If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address)
	■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
	■ Can be specified for any VSAM data set component if control interval processing is being used.
	■ Cannot be specified if the data set is being accessed through a path.
	■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.
TONUMBER(*num*)	For VSAM data sets only, code nm as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.

Remarks

Unprintable characters for 3270-type devices translate to periods before they display.

FIND formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, and DUMP. For load modules, the NUM, SNUM, and NONUM formats are equivalent to the LDUMP format and for VSAM data sets, the NUM, SNUM, and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time you enter a format operand on a LIST, FIND, or REPLACE subcommand, the value entered is used as the output format for subsequent LIST, FIND, and REPLACE subcommands. Also, note the following:

1 Formats NUM, SNUM, and NONUM limit the search and display length for logical records to 256 characters.

2 Column 72 is not searched for formats NUM or SNUM if the record format is fixed with 80 character records since this is the continuation column.

3 Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is performed). The other formats apply to logical records. For load modules, LBLOCK and LDUMP formats display only CSECT data.

TIP When using the `FIND` command against very large data sets (several MB), use the `BLOCK` parameter to avoid an ABEND.

4 Formats LBLOCK, LDUMP, BLOCK, and DUMP display only those segments of a record that contain the search string. A following segment also displays if the string spans a segment boundary.

5 For VSAM DATA or INDEX components, the LIST, FIND, and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

Control interval access is useful if a VSAM data set has logical errors. REPLACE repairs the error; however, since only the component is opened for update, the next access of the data set through the related cluster gets warning errors due to the differing time stamps.

The various FIND keywords are summarized by type below.

Keyword Type	Keywords
format	NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, DUMP
translate	CAPS, ASIS, IGNORE, PICTURE
position	OFFSET, MODULE, SKIPREC, MAXIN, MAXOUT, MAXFIND, SKIPCOL, MAXLEN
condition	THEN, ELSE
for source members	NUM, SNUM, NONUM, AND, ACOL, OR, OCOL, FORMAT
for load members	OFFSET, MODULE
for VSAM	FROMKEY, FROMADDRESS, FROMNUMBER, TOKEY, TOADDRESS, TONUMBER

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
----------------------- ISPMODE Session Display ------------  ROW 2891 OF 2946
COMMAND ===>                                                 ROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=PDS99T1* -----------------------
>----->fi pds99t1* / exec /

** FIND    PDS99T1A
000800 //ONE    EXEC  PGM=IEV90,REGION=2048K,
002400 //PDSAPPL  EXEC  PDSGEN,MEMBER=PDSAPPL
003200 //LK1    EXEC  PGM=IEWL,PARM='MAP,RENT,REUS,REFR,LET,NCAL',COND=(0,LT)
004400 //ABEND    EXEC PGM=IEFABEND,COND=(0,EQ)
PDS142I      43  lines in this member

** FIND    PDS99T1D
000800 //ONE    EXEC  PGM=IEV90,REGION=2048K,
002800 //*ARSE    EXEC  PDSGEN,MEMBER=@PARSE
003000 //DIALOG   EXEC  PDSGEN,MEMBER=@DIALOG
003200 //LK1    EXEC  PGM=IEWL,PARM='MAP,RENT,REUS,REFR,LET,NCAL',COND=(0,LT)
004400 //ABEND    EXEC PGM=IEFABEND,COND=(0,EQ)
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------- ISPMODE Session# 1 Log --------- ROW 1,000 TO 1,013 OF 1,013
COMMAND ===>                                                   SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=PDSDECRY  -------------------------
>----->find pdsdecry x4780c1x ldump

** FIND     PDSDECRY
PDS141I AT 000000  CSECT ENCRYPT   LENGTH  000015E0
  000180 0180  C1D9E240 5000F000  0A0612FF 4780C1A4  *ARS &.0.......Au*
  0001B0 01B0  30509180 40064780  C1D248E0 400406E0  *.&j. ...AK...
  0001E0 01E0  4780C1FC 48E04004  06E05810 400047F0  *..A..... ..0*

PDS141I AT 0017C8  CSECT R050A90    ENTRY DECIPH
  0018A8 0118  CB2C4740 C12A4780  C13259D0 CB304780  *... A...A......*
  0018B8 0128  C1324140 000247F0  C1364140 0001D237  *A.. ...0A.. ..K.*

PDS142I 7 blocks in this member
PDS146I 5 strings found
****************************** BOTTOM OF DATA ******************************
```

```
-------------------- ISPMODE Session# 1 Log# --------- Row 1,392 to 1,411 of 1,421
 COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=L*  -----------------------------
 >------>find l* 'P<<' picture

 ** FIND     LMDLIST
      Possible return codes are:
 PDS142I 125 lines in this member
 PDS146I 1 strings found

 ** FIND     LOGOUT
  D -    terminated StarTool.  After StarTool is terminated by CONDEND, the
  I -      HISTORY  member  MODULE({* / Fullm / Partm*})
  D -              MODULE({* / Fullm / Partm*})
  RN-                LKED(Partl)
  RO-              LKED(Partl)
  I -                TRANS(Partt)
  D -              TRANS(Partt/ ASM      / ASMA90   / ASMH     / COBOL    /
  RN-                USERDATA(Partu) / SYSMOD(Partu)
  RO-              USERDATA(Partu) / SYSMOD(Partu)
  RN-                ZAP(Partz)
  RO-              ZAP(Partz)
  D -                 Note that if MODULE(Fullm) or TRANS(Partt) is also
```

```
FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
--------------------- ISPMODE Session# 1 Log ----------ROW 1,000 TO 1,017 OF 1,017
COMMAND ===>                                                    ROLL ===> CSR
- DSN=SER07.VSAM.AX.CLUSTER.PATH,VOL=SER=SER007  ----------------------------
>----->find 'vsam' dump
PDS140I DUMP RECORD      23    LENGTH  7,623   RBA 00024576
0007MM F+0010  40C9C4C3 F3F3F5F1  C9405C5C 40E5E2C1  * IDC3351I ** VSA*
0007MM F+0020  D440C040 D6D7C5D5  404F40C3 D3D6E2C5  *M  OPEN | CLOSE*
0007MM F+0190  99899587 40E5E2C1  D4409799 968385A2  *ring VSAM proces*
0007MM F+08A0  99969940 81838385  A2A28995 8740E5E2  *ror accessing VS*
0007MM F+08B0  C1D440A5 9693A494  85408481 A381A285  *AM volume datase*

PDS140I DUMP RECORD      27    LENGTH  1,089   RBA 00013377
000730  +0020  99408183 8385A2A2  89958740 E5E2C1D4  *r accessing VSAM*

PDS142I 80 blocks in this data set
PDS146I 4 strings found
****************************** BOTTOM OF DATA ******************************
```

FINDMOD Subcommand

The FINDMOD subcommand searches for all copies of a system routine.

Example
```
FINDMOD   iefbr14
```

Syntax
```
FINDMOD module [SYSTEM|NUCLEUS|ADDRESS]
               [CHANGE|GO  LIB(num)    ]
               [NOSEARCH              ]
```

Aliases
FINDM, FINDMO, FINDMOD

Defaults

SYSTEM, LIB(1)

Required

module

Operands

module	Identifies the member to be found.
SYSTEM	Issue a BLDL and search the LPA and MLPA. If the member is in MLPA, TASKLIB, or LINKLIST, search the LINKLIST concatenation. If the member is in LPA, search the LPALIB concatenation. Also, search the in-storage nucleus.
NUCLEUS	The search should take place in the in-storage nucleus.
ADDRESS	Module is an address whose corresponding routine is to be located in the LPA, MLPA, or NUCLEUS. Search the LINKLIST libraries if the module is in MLPA and search the LPALIB concatenation if it is in LPA.
CHANGE	Specifies that if the member is found, perform a CHANGE to that library.
GO	Specifies that if the member is found, initiate a GO session for that library.
LIB(n)	If there are multiple finds, LIB specifies which library to select for CHANGE or GO. If you specify a higher number for LIB than actually found, the last data set is selected.
NOSEARCH	Specifies that the LINKLIST and LPALIB library concatenations are not to be searched.

Remarks

If FINDMOD is entered with the SYSTEM parameter (the default) the following search takes place:

1   A BLDL is issued and reported on.

2   The LPA and MLPA is searched (the module may be in both).

3   If the module is in MLPA, TASKLIB, or the LINKLIST, the LINKLIST libraries are individually searched.

4   If in LPA, the LPALIB libraries are individually searched.

5   The in-storage nucleus map is searched.

If FINDMOD is entered with the ADDRESS parameter, the module parameter is assumed to be a hex address and the following search takes place:

1   The in-storage nucleus map is searched.

2   If in MLPA, the LINKLIST libraries are individually searched.

3   If in LPA, the LPALIB libraries are individually searched.

If FINDMOD is entered with the NUCLEUS parameter, only the in-storage nucleus map is searched.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------- ISPMODE Session# 1 Log ---------- ROW 1,000 TO 1,015 OF 1,015
COMMAND ===>                                                    SCROLL ===> CSR
- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011  -----------------------
>----->findmod igx00011 change
PDS293I Member found in LINKLIST DSNAME='ISF.V1R3M2.ISFLPA'
PDS294I Member found in MLPA
PDS295I Address:02A13DD8; length:000200
PDS295I Attributes 1: Reentrant; Reusable; Not loadable only;
PDF295I Attributes 2: Authorized library; Authorized program; Amode 64;
PDF295I Attribute 3: Resident above 16Meg; PLPA resident;

Change 'ISF.V1R3M2.ISFLPA'
PDS200I DISP UNIT    RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390    U        0   6160    1X      6       5       0 TRK       4

PDS222I Block allocation: SPACE=(6160,(40,,5))

PDS223I This is a linklist data set

PDS224I This data set is APF authorized

PDS300A ENTER OPTION -- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011
****************************** BOTTOM OF DATA ******************************
```

FINDPROC Subcommand

The FINDPROC subcommand searches the active JES procedure library list and returns the names of the libraries that contain the specified JCL procedure.

Example

FINDPROC $AUTH LIST

returns the following:

```
   PROCLIB(0001) DSN=SYS1.SERENA.TCPIP19.PROCLIB
   PROCLIB(0002) DSN=SYS1.SERENA.PROCLIB
PDS293I Member found in PROCLIB(0002) DSNAME='SYS1.SERENA.PROCLIB'
   PROCLIB(0003) DSN=SYS1.D002.PROCLIB
   PROCLIB(0004) DSN=SYS1.CPZOS21.BLDJ.PROCLIB
   PROCLIB(0005) DSN=SYS1.IBM.BLDJ.PROCLIB
   PROCLIB(0006) DSN=ADCD.ZOSV1R3.PROCLIB
   PROCLIB(0007) DSN=CEE.SCEEPROC
   PROCLIB(0008) DSN=CSQ520.SCSQPROC
   PROCLIB(0009) DSN=IOE.SIOEPROC
   PROCLIB(0010) DSN=EOY.SEOYPROC
   PROCLIB(0011) DSN=HLA.SASMSAM1
   PROCLIB(0012) DSN=CBC.SCCNPRC
   PROCLIB(0013) DSN=IGY310.SVSC.PROCLIB
   PROCLIB(0014) DSN=IEL111.SVSC.PROCLIB
   PROCLIB(0015) DSN=FAN130.SVSC.PROCLIB
   PROCLIB(0016) DSN=SYS1.PROCLIB
   PROCLIB(0017) DSN=USER.D002.PROCLIB
   PROCLIB(0018) DSN=USER.PROCLIB
```

Syntax

```
FINDPROC module [LIST]
                [CHANGE|GO LIB(n)]
```

Aliases

```
FINDP,FINDPR,FINDPRO
```

Defaults

```
LIB(1)
```

Operands

module	Name of the target JCL procedure member.
LIST	Lists the procedure libraries that are searched.
CHANGE	If the member is found, perform a CHANGE to that library.
GO	If the member is found, initiate a GO session for that library.
LIB(*n*)	If there are multiple finds, LIB specifies which library to select for the CHANGE or GO action. If you specify a higher number for LIB than the number actually found, the last data set is selected.

NOTE As part of FINDPROC processing, FDM creates and activates console SZFCONSL to issue the $D PROCLIB command, but then inactivates it.

FIXPDS Subcommand

The FIXPDS subcommand alters attributes of the current data set. It contains functions for sequential, PDS, and PDSE data sets. This subcommand also automatically invokes DSNAME.

Example

```
FIXPDS   RELEASE RECFM(FBA)
```

Syntax

```
FIXPDS
        [NEWDSNAME(new.data.set.name)]
        [MODDSNMAME(data.set.to.modify) MODVOLUME(volume)]¹

        [ORDER]
        [EXPANDDIR(num_add)|FREEDIR(num_unused)|
           DIRFREE(percent)|INITDIR(rblks)|
          {RESET(tblks)|RESET¹|
               [CHECK¹|NOCHECK¹] }]
        [RELEASE¹|RELEXTENT|RELSAVE(trks)|RELFREE(percent)|MAXSPACE]
        [ADDTRK(trks)|ADDCYL(cyls)|ADDFREE(percent)]
        [DSCB(offset, hexdata_to_verify, hexdata_to_replace)]¹
        [*TRK  |  *CYL  |  *BLK [*ROUND]                      ]
        [*SPACE(number)                                       ]
        [*EXPDT(yyddd) | *RETPD(nnnn)                         ]
        [*MGMTCLAS(mgmtclas)                                  ]
        [*STORTCLAS(storclas)                                 ]
        [ASM | CLIST | CNTL | LOAD | OBJECT                   ]
        [DSORG(PS|PSU | DA|DAU | PO|POU | IS|ISU | VS|VSU)    ]
        [RECFM(F|FB|FA|FM|FBA|FBM|V|VB|VA|VM|VBA|VBM|U|UA|UM)]
        [LRECL(L1)                                            ]
        [*BLKSIZE(L2)                                         ]
        [*OPTCD(W|WC|WCZ|C|CZ|Z) | *NOOPTCD                   ]
```

¹NOTE: Supported for PDSE data sets.

Aliases

FIX, FIXP, FIXPD

Defaults

NOCHECK, RECFM(previous value), LRECL(previous value), BLKSIZE(previous value), OPTCD(previous value)

Required

Operands

EXPANDDIR(#add)	Expands the directory by adding #add unused directory blocks. Type #add as a number from 0 through 99999. The default is 50.
FREEDIR(#unused)	Expands or reduces the directory as necessary to obtain #unused free directory blocks. Type #unused as a number from 0 through 99999. It defaults to 5.
ORDER	Reads the directory looking for members that are duplicates or out of order. Each such member causes a prompting sequence before any fix is attempted. For out of order members, the member is renamed back to itself in two steps so that it can be placed in collating sequence. For duplicate member names, one of the members is renamed (to a name low in the collating sequence). Dispose of either one or both later.
RESET(Totblks)	Writes over the current directory with empty directory blocks. Enter totblks as a number from 1 through 99999. It defaults to the number of directory blocks already allocated. After the directory blocks are written, the end of data marker (DS1LSTAR) is reset to indicate an empty partitioned data set. This is the equivalent of a COMPRESS for an empty partitioned data set.
RESET	Reset the data set to an empty data set.
INITDIR(rblks)	Writes over the current directory with empty directory blocks. Enter rblks as a number from 1 through 99999. It defaults to the number of directory blocks already allocated. The data set end of data marker (DS1LSTAR) is not modified as for RESET.
CHECK	Check each member to see if it is being edited for options RESET or INITDIR.
NOCHECK	No edit check is desired.
DIRFREE(perc)	Specifies the percent of free directory blocks desired (0 to 999). For example, if 20 is entered for a data set with 50 used directory blocks, the number of free directory blocks will be adjusted (up or down) to 10.
ADDFREE(perc)	Specifies the percent of unused space to be available in a data set (0 to 999). For example, if 20 is entered for a data set with 50 used tracks and 5 free tracks, another 5 tracks will be added to the data set in track or cylinder increments. If RELFREE and ADDFREE are both used, ADDFREE is performed first.
RELFREE(perc)	Specifies the percent of unused space to be retained in a data set (0 to 999). For example, if 20 is entered for a data set with 50 used tracks, any unused space past track 60 will be returned to the system. If the data set was allocated with cylinder secondary space, any returned disk space will maintain cylinder boundaries for the data set. If RELFREE and ADDFREE are both used, ADDFREE is performed first.

RELEASE	Returns any unused disk space in the current data set to the system. If the data set was allocated with cylinder secondary space, any returned disk space will maintain cylinder boundaries for the data set.
RELEXTENT	Returns any unused disk extents to the system.
RELSAVE(nnn)	Returns any unused disk space in the current data set to the system except for the first nnn tracks. If the data set was allocated with cylinder secondary space, any returned disk space will maintain cylinder boundaries for the data set.
MAXSPACE	Marks all extents of the data set as "in-use" by adjusting the end of data set marker (DS1LSTAR). This could be used after a data set compress to RESTORE members after the used portion of the data set.
ADDTRK(trka)	Adds a single secondary extent of trka tracks to the data set. This operation is independent of secondary space indicators from the data set label.
ADDCYL(cyla)	Adds a single secondary extent of cyla cylinders to the data set. This operation is independent of secondary space indicators from the data set label.
DSCB(offs,ver,rep)	Modify the Format 1 DSCB as specified by the OFFS, VER, and REP parameters. The DSCB keyword may contain multiple sets of OFFS, VER, and REP. For example, the following subcommand changes the first character of DS1DSSN and sets DS1EXPDT to zero if it is set to any 1995 value: **FIXPDS DSCB(2D C1C2C3C4C5C6 F1 38 5F 000000)** Each OFFSET may vary from 2C to 8C. Each VER must contain 2 to 20 hexadecimal characters that are checked against the corresponding DSCB data. If any VER fails, the DSCB keyword has no effect. Each REP must contain 2 to 20 hexadecimal characters that are used to update corresponding DSCB data. If all sets of OFFS, VER and REP are processed successfully and you confirm that these changes are desirable, DSCB updates are performed by PDSEAUTH.
NEWDSNAME(dsname)	Specifies a new data set name for an uncataloged data set on a non-indexed volume. If the new DSNAME begins with 'NULL..' the data set is also deleted. If you are authorized to use this facility, Format 1 DSCB updates are made with PDSEAUTH. This facility can rename a data set even if it is ENQUEUED.
MODDSNAME(dsname)	Specifies a data set whose Format 1 DSCB is to be modified. If you are authorized to use this facility, DSCB updates are made with PDSEAUTH. Use this facility to repair damaged data sets even if they cannot be allocated directly.
MODVOLUME(volume)	A volume name for use with the MODDSNAME keyword if the data set is not cataloged or is mis-cataloged.

TRK	Change the secondary space allocation type to tracks. Indicate the number of tracks with a SPACE keyword. This keyword modifies the Format 1 data set control block.
CYL	Change the secondary space allocation type to cylinders. Indicate the number of cylinders with a SPACE keyword. This keyword modifies the Format 1 data set control block.
BLK	Change the secondary space allocation type to block allocation. Change the number of blocks with a SPACE keyword. Specify the ROUND keyword to get space in cylinder boundaries. This keyword modifies the Format 1 data set control block.
ROUND	Specified with BLK to indicate that secondary space be obtained on cylinder boundaries. This keyword modifies the Format 1 data set control block.
SPACE(number)	Usually specified with TRK, CYL, or BLK to indicate the number of units to obtain for a secondary allocation. This keyword modifies the Format 1 data set control block.
EXPDT(yyddd)	Specifies the expiration date for the data set. Delete or write over the data set on or after the expiration date. Use the format EXPDT(yyyyddd). This keyword modifies the Format 1 data set control block; however, if the data set is SMS managed, IDCAMS is invoked to update this field in the catalog and the Format 1 data set control block.
RETPD(nnnn)	Specifies the number of days (0 to 9999) to retain the data set.
MGMTCLAS(mgmtclas)	For SMS-managed data sets, invokes IDCAMS to update the management class of a data set.
STORCLAS(storclas)	For SMS-managed data sets, invokes IDCAMS to update the storage class of a data set.
ASM	Modifies the DCB to RECFM(FB) LRECL(80) and the BLKSIZE shown in the table below. RECFM, LRECL, or BLKSIZE may be explicitly specified.
CLIST	Modifies the DCB to RECFM(VB) LRECL(255) and the BLKSIZE shown in the table below. RECFM, LRECL, or BLKSIZE may be explicitly specified.
CNTL	Modifies the DCB to RECFM(FB) LRECL(80) and the BLKSIZE shown in the table below. RECFM, LRECL, or BLKSIZE may be explicitly specified.
LOAD	Modifies the DCB to RECFM(U) and the BLKSIZE shown in the table below. RECFM or BLKSIZE may be explicitly specified.
OBJECT	Modifies the DCB to RECFM(FB) LRECL(80) and the BLKSIZE shown in the table below. RECFM, LRECL, or BLKSIZE may be explicitly specified.

FIXPDS BLKSIZE and device table

	3350	3375	3380	3390	9345
ASM block sizes	6160	6800	6320	6480	7200
CLIST block sizes	6232	6816	6356	6518	6088
CNTL block sizes	6160	6800	6320	6480	7200
LOAD block sizes	19069	32760	32760	32760	32760
OBJECT block sizes	2960	3200	3120	3120	3200

DSORG(Newd)	The desired data set organization for the data set. Normally, the DSORG of a data set would only be changed to add or remove the **U** (Unmoveable) attribute; however, the DSORG of DA and PS type data sets are sometimes also manipulated. The new DSORG may be any of the following: PS, PSU, DA, DAU, PO, POU, VS, VSU, IS, or ISU. Changing this parameter does not change the data set physically The new value is just changed in the data set label (Format 1 DSCB).
RECFM(Newr)	A new DCB RECFM value for the data set. Newr defaults to its previous value for the data set or may be entered as any one of the following: F, FB, FA, FBA, FM, FBM, V, VB, VA, VBA, VM, VBM, U, UA, or UM. Changing this parameter does not change the data set physically. The new value is just changed in the data set label (Format 1 DSCB).
LRECL(L1)	A new DCB LRECL value for the data set. L1 defaults to its previous value for the data set or may be entered as a number from 1 through 32767. Changing this parameter does not change the data set physically. The new value is just changed in the data set label (Format 1 DSCB).
BLKSIZE(L2)	A new DCB BLKSIZE value for the data set. L2 defaults to its previous value for the data set or may be entered as a number from 1 through 32767. Changing this parameter does not change the data set physically. The new value is just changed in the data set label (Format 1 DSCB).
OPTCD(Newo)	The desired OPTCD value for the data set. Newo defaults to its previous value for the data set or may be changed to any of the following: W, WC, WCZ, C, CZ, or Z.
NOOPTCD	Turn off any current W, C, or Z OPTCD value.

Remarks

After a FIXPDS subcommand gains control, it writes messages regarding the data set action to be taken and any potential error conditions. FIXPDS prompts you before making any data set changes unless CONTROL NOPROMPT is in effect. If you invoke the FIXPDS subcommand with no operands in an ISPF environment, StarTool FDM prompts with a panel appropriate for the type of data set you are currently processing.

The following are more detailed logic notes on several of the different functions available through the FIXPDS subcommand.

EXPANDDIR (expand the directory—ignored for sequential) or FREEDIR (adjust free blocks—ignored for sequential)	■ If no members exist in the data set, RESET is used ■ The names of members that will be moved because of the expanded directory are listed ■ If any of these members are in use by an ISPF edit session, a warning message is issued ■ As each member is moved, an informational message is issued ■ The additional directory blocks are written
RESET (reset the directory or the data set)	■ If any members exist in the data set, a warning message is issued ■ If CHECK is specified and any members are in use by an ISPF edit session, a warning message is issued ■ The data set's directory blocks are rewritten with null data, destroying any existing member information. For a PDSE, a STOW with "initialize" deletes all members ■ The end of data marker (DS1LSTAR) is reset to indicate an empty data set (this is not performed on a PDSE) ■ A PDSE must be allocated as DISP=OLD prior to performing the FIXPDS RESET function. This can be done with the following command: `CH  pdse.dsn  OLD`

INITDIR (initialize the directory — ignored for sequential)	■ If any members exist in the data set, a warning message is issued ■ If CHECK is specified and any members are in use by an ISPF edit session, a warning message is issued ■ The data set's directory blocks are rewritten with null data, destroying any existing member information ■ The end of data marker (DS1LSTAR) is *not* modified
RECFM, LRECL, BLKSIZE (modify the DCB attributes — a warning is issued)	■ If a data type keyword such as ASM, CLIST, CNTL, LOAD, or OBJECT is entered, RECFM and LRECL default values are provided from a table and a BLKSIZE value is determined based on the allocated device type ■ If any RECFM, LRECL, or BLKSIZE values are specifically requested by keyword, these values will override the values as determined from the data type keywords entered ■ The data set is opened for output with these new DCB parameters
RELEASE (return space to operating system)	■ If the data set is currently open, an error message is issued ■ The amount of space and the type of units (TRK or CYL) is determined ■ The data set is opened with EXTEND to get the space released

MAXSPACE (mark all allocated space in use)	All extents of the data set are marked in use by setting the end of data marker (DS1LSTAR) value to the last disk address for the data set. Use this with a RESTORE subcommand to recover members that were deleted prior to a data set compress. It may also be possible to resurrect members lost from a PDS after the directory is overwritten by sequential output due to some sort of JCL or procedural error. The sequential data blocks take one or more tracks followed by the end of file marker in the disk tracks formerly occupied by the directory. To attempt recovery of PDS members located after the overwritten tracks, enter:
	■ FIXPDS MAXSPACE (to mark all allocated space in use)
	■ FIXPDS INITDIR(43) ADDTRK(1) (to reinitialize the directory and add another extent)
	■ RESTORE name REPEAT NOPROMPT NODISPLAY (to restore all deleted members)
	■ VERIFY : (to find invalid members after the previous DS1LSTAR)
	■ BROWSE name (to scan contents of members)
	■ RENAME name realname (to name needed members properly)
	■ DELETE name (to delete members not needed)
	■ COMPRESS (to clean up unused space again)

ADDTRK or ADDCYL (add one secondary extent)	■ An extent is added to the data set in the units given in the keyword. For example, ADDTRK(12) adds one secondary extent of 12 tracks while ADDCYL(4) adds one secondary extent of 4 cylinders
	■ This operation is independent of any secondary extent information associated with the data set. It can add a single extent of the specified size to a data set even if the data set was allocated without secondary extents.
	■ StarTool FDM issues a warning message if you attempt to add a secondary extent to any data set named 'xxx.NUCLEUS' (where xxx is any dsname prefix). Secondary extents are not supported for the 'SYS1.NUCLEUS' data set. If this data set were used as an operating system NUCLEUS data set, you could not IPL!
	■ StarTool FDM also issues a warning message if you attempt to add an extent to a LINKLIST data set. While this is supported by the operating system, the extent would not be available for use until the next system IPL.

| NEWDSNAME (rename), MODDSNAME, and DSCB (direct DSCB update)

 From the FIXPDS panel, select "Examine or update DSCB". | ■   A prompt panel displays showing current DSCB value. This panel contains field-level help to provide guidance.If any changes are made in this panel, a FIXPDS DSCB subcommand will be generated

 ■   The DSCB keyword may contain multiple triples of hexadecimal information with values for: OFFSET VER REP

 ■   As an example, the following subcommand changes the first character of DS1DSSN and sets DS1EXPDT to zero if it is set to any 1995 value:

       FIXPDS DSCB(2D C1C2C3C4C5C6 F1 38 5F 000000)

 ■   Each OFFSET may vary from 2C to 8C

 ■   Each VER must contain 2 to 20 hexadecimal characters which are checked against the corresponding DSCB data

 ■   If any VER fails, the DSCB keyword has no effect

 ■   Each REP must contain 2 to 20 hexadecimal characters which are used to update corresponding DSCB data

 ■   If all sets of OFFS, VER, and REP are processed without problems and you confirm that these changes are desirable, DSCB updates will be performed by PDSEAUTH

 ■   Since DSCB operates on a conditional basis, it can be set up to operate on a global basis where only matching data sets are modified, as in the following example:

 `GLOBAL FIXPDS DSCB(2D C1C2C3C4C5C6 F1 38 5F 000000)`

 ■   If you lower the BLKSIZE of a data set with FIXPDS, this only changes the data set label. You need to find all members with a larger BLKSIZE using VERIFY :. In a source library, reconstruct these members with the REPRO subcommand

 This subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection". |

```
------------------- ISPMODE Session Display ------------------  ROW 2957 OF 2970
COMMAND ===>                                               SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=PDS99T1* ------------------------
>----->fix freedir(45)
PDS050I TESTMSG will be moved
PDS050I TMP347 will be moved

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80    9040    1X    47      10    40 TRK      25

PDS392A Should this data set be modified (Yes/No) ?
>----->y
PDS051I TESTMSG is being moved
PDS051I TMP347 is being moved


PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80    9040    1X    47      10    40 TRK      45
```

```
------------------------ ISPMODE Session# 2 Log# 1 --  ROW 305 TO 315 OF 315
COMMAND ===>                                               SCROLL ===> CSR
- DSN=SER07.LIB.CLISTV,VOL=SER=SER006  MEM=TRAP  ----------------------------
>----->fix addfree(50)

PDS200I DISP UNIT     RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390     VB     255   32760    1X    10       0     3 TRK       1

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

PDS200I DISP UNIT     RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390     VB     255   32760    2X    15       5     3 TRK       1

****************************** BOTTOM OF DATA ******************************
```

FSE Subcommand

The FSE subcommand edits a member. Use the TSO FSE command.

Example

```
FSE   mema:memb cntl
```

Syntax

```
FSE   memgroup [ASIS
               [OLD|NEW
               [NONUM
               [ASM|BASIC|CLIST|CNTL|COBOL|DATA|
                FORTE|FORTG|FORTGI|FORTH|GOFORT|
                IPLI|LIST|PLI|PLIF|TEXT|VSBASIC ]
```

Aliases

```
FS, FSE
```

Defaults

memgroup, EDIT type based on the data set name

Required

Operands

memgroup	Identifies the members to be edited.
	Default member names, member lists, member name range, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
ASIS	Edits the member with upper- and lower-case characters.
OLD	Verifies that the member exists before calling FSE.
NEW	Verifies that the member does not exist before calling FSE.
NONUM	Edits the member without using line numbers.
ASM	EDIT type is assembly.
BASIC	EDIT type is BASIC.
CLIST	EDIT type is CLIST.
CNTL	EDIT type is CNTL.
COBOL	EDIT type is COBOL.
DATA	EDIT type is DATA.
FORTE	EDIT type is FORTRAN level "E".
FORTG	EDIT type is FORTRAN level "G".
FORTGI	EDIT type is FORTRAN level "GI".
FORTH	EDIT type is FORTRAN level "H".
GOFORT	EDIT type is GOFORT.
IPLI	EDIT type is IPLI.
LIST	EDIT type is LIST.
PLI	EDIT type is PLI.
PLIF	EDIT type is PLI (F-level).
TEXT	EDIT type is TEXT.
VSBASIC	EDIT type is VSBASIC.

Remarks

If you enter an EDIT type keyword, that keyword is passed to FSE as the descriptive qualifier; otherwise, a descriptive qualifier is chosen as follows:

1 If the low-level qualifier of the partitioned data set name is one of the valid descriptive qualifiers for FSE (ASM, BASIC, CLIST, CNTL, COBOL, DATA, IPLI, LIST, PLI, TEXT, or VSBASIC), that qualifier is passed to FSE.

2 If the low-level qualifier is FORT, GOFORT is passed to FSE.

3 Otherwise, the data type qualifier is not a valid descriptive qualifier and the general descriptive qualifier, DATA, is passed to FSE.

This interface is optional, and should only be used if you have FSE+ installed at your installation.

GO Function

The GO command allows you to initiate or switch to parallel StarTool FDM sessions.

Example

```
GO   'sys1.parmlib'
```

Syntax

```
GO [* | ** | gonum | dsname [VOLUME(volser)] [SHR|OLD]
      |   FILE(ddname) [NUMBER(num)]
   [REPLACE                            ]
```

Aliases

```
G, GO
```

Defaults

SHR or previously used GO data set if an ***** is entered.

Required

Operands

*	Switches to the most recently used GO session. Use this to toggle GO sessions. This is similar to the toggling for CHANGE *.
**	Rotates through active GO sessions. Set a PF key to GO **.
gonum	The number (1 through 9) of the parallel session to initiate or switch to.
dsname	The data set name for the parallel session to initiate or switch to. If the data set name is not entered in single quotes ('), your TSO PREFIX is appended to the start of the entered data set name.
VOLUME(volser)	The volume name to use for uncataloged data sets.
SHR	Allocates the data set with a disposition of SHR; allows simultaneous use of this data set by others. The use of SHR is recommended.
OLD	Allocates the data set with a disposition of OLD; does not allow simultaneous use of this data set by others. The use of SHR is recommended.
FILE(ddname)	Identifies the DDNAME of a pre-allocated data set. Only disk data sets (including VIO) are supported. If you use the FILE keyword, then dsname, SHR/OLD, VOLSET, and VOLUME should not also be used. If the data set is concatenated, use the SHR or OLD keyword since StarTool FDM reallocates the data set.
NUMBER(num)	The concatenation number desired for the DDNAME allocation for the FILE keyword. *num* defaults to 1 but if *num* is larger than the number of concatenated data sets, the last data set in the sequence is used.
REPLACE	Replaces a session (this is when a session number and a data set name are both specified).

Remarks

The initial session invoked by StarTool FDM is called GO session 1. To establish an alternate parallel session, enter GO and a session number or a data set name. StarTool FDM prompts for the data set name, if necessary. A parallel session is initiated. To transfer to an established parallel session, enter GO and the session number or data set name.

When you are in an alternate session, the log is shared between sessions. MEMLIST, CSECTS, and ZAP processing is maintained uniquely between the different sessions. The other functions are shared between sessions.

Delete a GO session with the DROP command. The syntax is:

```
DROP  [*/num]  [PROMPT]
```

HELP Subcommand

The HELP subcommand provides information for using StarTool FDM. HELP operates differently depending on your operating mode. For example, if you are in ISPMODE or MEMLIST, HELP provides standard ISPF tutorial information. In line mode or batch, the HELP subcommand lists data from the HELP data set.

Example

```
HELP
```

Syntax

```
HELP   [subcommand [FUNCTION] [SYNTAX] [OPERAND|OPERAND(k)]]
```

```
Note: the above operands are only defined in line mode or batch.
```

Aliases

```
H, HE, HEL, HELP
```

Defaults

Required

Operands

subcommand	The name of a subcommand (or the alias name of a subcommand) for which HELP is to be displayed. If this operand is omitted, a list of valid subcommands displays.
FUNCTION	Display function information for the subcommand.
SYNTAX	Display syntax information for the subcommand.
OPERAND(k)	Display information for the keyword identified in the (k) operand. If the (k) operand is omitted, information on all operands of the subcommand displays.

Remarks

The HELP subcommand provides information on using StarTool FDM.

Use either the TSO HELP or HEL command. This choice is made during StarTool FDM installation. To determine which program is used, enter a CONTROL DEFAULT subcommand and look for an output line beginning "HELP calls".

This interface is optional. Use it only with the HELP command processor or the public domain HEL command.

HEX Function

The HEX command performs hexadecimal calculations.

Example
```
HEX
```

Syntax
```
HEX
```

Aliases
```
HEX
```

Operands

No operands are supported for the HEX subcommand.

Remarks

The HEX ISPF table displays in response to a HEX command. When you are in a HEX display, you can delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, etc. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as a primary command.

The HEX table displays the calculator tape for your calculator. Enter new calculations from the HEX display.

The following primary commands are supported directly for the HEX function. For information on ISPMODE commands available in StarTool FDM, see <Xref>"Common Commands" in Chapter 2, "ISPF Interface Commands".

EDITT[BL] (or ET[BL])	Enters an edit session on HEX table data.	
F	Finds a string and positions the display start location.	
	Syntax	`F anystring [ASIS]` `    [FIRST|LAST|PREV][PREFIX|SUFFIX|WORD]`
OUT[PUT]	Outputs the HEX table to print or a data set.	
	Syntax	`OUTPUT [=c | F(ddname)]`

REM[OVE]	Trims the HEX table based on a string match.	
	Syntax `REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
X	Clears the HEX table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the HEX table; this is equivalent to X ALL	

HISTORY Subcommand

The `HISTORY` subcommand displays the last linkage edit date for a module and lists any CSECT IDR data assigned to that load module.

Example

```
HISTORY    mema:memb
```

Syntax

```
HISTORY        memgroup
               [ML|MEMLIST|NEWML|SUBLIST]
               [MODULE({*|Fullname|Part*}) {ASIS|UPPERCASE|UC}]
               [COBOL(ADV|NOADV,              APOST|NOAPOST,
                   AWO|NOAWO,                 CMPR2|NOCMPR2,
                   COUNT|NOCOUNT,             COMPCLASS|COMPPROGRAM,
                   CURRENCY|NOCURRENCY,       DATA24|NODATA24,
                   DATA31|NODATA31,           DATEPROC|NODATEPROC,
                   DBCS|NODBCS,               DECK|NODECK,
                   DUMP|NODUMP,               DYNAM|NODYNAM,
                   ENDJOB|NOENDJOB,           EVENTS|NOEVENTS,
                   FASTSRT|NOFASTSRT,         FDUMP|NOFDUMP,
                   INTDATELIL|INTDATEANSI,    FLOW|NOFLOW,
                   LIB|NOLIB,                 LIST|NOLIST,
                   MAP|NOMAP,                 NAME|NONAME,
                   NUMBER|NONUMBER,           NUMCLSALT|NUMCLSPRIM,
                   NUMPROCMIG|NONUMPROCMIG,   NUMPROCPFD|NONUMPROCPFD,
                   OBJECT|NOOBJECT,           OBJ370|NOOBJ370,
                   OFFSET|NOOFFSET,           OPTFULL|NOOPT,
                   OPTIMIZE|NOOPTIMIZE,       OUTDD|NOOUTDD,
                   PGMLONGMIX|NOPGMLONGMIX,   PGMLONGUPP|NOPGMLONGUPP,
                   QUOTE|NOQUOTE,             READYTRACE|NOREADYTRACE,
                   RENT|NORENT,               RESIDENT|NORESIDENT,
                   RMODEANY|RMODE24,          SEQUENCE|NOSEQUENCE,
                   SIZEMAX|NOSIZEMAX,         SOURCE|NOSOURCE,
                   SSRANGE|NOSSRANGE,         STATE|NOSTATE,
                   SYMDMP|NOSYMDMP,           TERMINAL|NOTERMINAL,
                   TEST|NOTEST,               TESTBLOCK|NOTESTBLOCK,
                   TESTPATH|NOTESTPATH,       TESTSTMT|NOTESTSTMT,
                   TRUNCBIN|TRUNCBIN,         TRUNCSTD|NOTRUNCSTD,
                   VBREF|NOVBREF,             WORD|NOWORD,
                   XREF|NOXREF,               ZWB|NOZWB)]
               [EXTERN|WKEXTERN|PSEUDOREG|LABELREF|COMMON|PRIVATE|CSECT]
               [GENERATE|CMNLOAD]
               [ALL|
                   {LKED(Lpartname)|NOLKED}
                   {TRANS(Tpartname|ASM|ASMA90|ASMH|C|C++ZOS|C370|COBOL|
                       COBOLII|COBOLMVS|COBOLOS|COBOLVS|COBOLV4|COBOLZOS|
                       FORTRAN|FORTRANG|FORTRANH|FORTRANV|IFOX|PASCAL|
                       PLS|PL/X-390|PL1|PLX370|REXX370|RPG|SASC)|NOTRANS}
                   {ZAP(Zpartname)|NOZAP}]
               [USERDATA(Upartname)|SYSMOD(Upartname)|NOUSERDATA|NOSYSMOD]
               [NOSYSTEM]
```

Aliases

```
HI, HIS, HIST, HISTO, HISTOR, HISTORY
```

Defaults

memgroup, UPPERCASE, ALL, TRANS (from CONTROL), LKED, ZAP, and USERDATA. If LKED, TRANS, ZAP, or USERDATA are specified, only the corresponding types of data are provided.

Required

Operands

memgroup	Identifies the load members for which you want history data, using standard MEMGROUP syntax.
	Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
`MODULE(`*name*`)`	Specifies a 1- to 8-byte partial external name that limits CSECT names for history reporting. If you enter this parameter with any TRANS SYSMOD/USERDATA or ZAP keywords, only those CSECTs that satisfy the MODULE keyword are checked for the other keyword conditions.
	The MODULE operand takes the following forms:
	■ MODULE(*) – uses name entered in the previous MODULE option
	■ MODULE(*Fullname*) – reports on CSECTs with this name only
	■ MODULE(*Part**) – reports on CSECTs with names conforming to this wildcard pattern
`ASIS|UPPERCASE`	Lets the user specify how the `MODULE(`*name*`)` parameter is processed. This parameter must immediately follow the MODULE parameter in the parameter list. It takes one of the following values:
	`ASIS` Case is respected as typed for the CSECT name in the `MODULE(`*name*`)` parameter.
	`UPPERCASE` The CSECT name in the `MODULE(`*name*`)` parameter is converted to uppercase characters before executing the `HISTORY` command. This is the default value. The alias `UC` is accepted for this value.
	Customers using C++ translators and customers with lower-case CSECT names in their code should use the `ASIS` parameter with the `HISTORY` command.

COBOL(opt1, opt2, opt3, ...)	Specifies one or more COBOL compiler options that are to be matched for a COBOL CSECT. If all options entered for the given compiler type match for a CSECT, the CSECT is listed.
	If you also enter MODULE or TRANS parameters, the COBOL keyword applies only to those CSECTs that pass the filtering.
	Options for the COBOL parameter vary by translater used. Supported options are listed by translater below.
	All COBOL Compilers:
	■ OPTIMIZE|NOOPTIMIZE – optimize object code
	■ TEST|NOTEST – object code for debug
	COBOL II, COBOL for MVS and VM, COBOL for OS/390 and VM, and COBOL for z/OS:
	■ ADV|NOADV – byte for printer control character
	■ APOST|NOAPOST – apostrophe (') is non-numeric delimiter
	■ AWO|NOAWO – APPLY WRITE-ONLY clause
	■ CMPR2|NOCMPR2 – COBOL II release 2 compatible code
	■ COMPCLASS|COMPPROGRAM – (not COBOL II) compile unit is class
	■ CURRENCY|NOCURRENCY – (not COBOL II) alternate symbol for $
	■ DATA24|NODATA24 – dynamic storage is below 16 Megabytes
	■ DATA31|NODATA31 – dynamic storage is unrestricted
	■ DATEPROC|NODATEPROC – (not COBOL II) unknown at present
	■ DBCS|NODBCS – Double Byte Character Set shift codes
	■ DECK|NODECK – object code output to //SYSPUNCH
	■ DUMP|NODUMP – system dump if the compiler ABENDs
	■ DYNAM|NODYNAM – CALLed programs invoked dynamically
	■ EVENTS|NOEVENTS – (not COBOL II) messages to SYSEVENT
	■ FASTSRT|NOFASTSRT – invoke external product to perform sort
	■ FDUMP|NOFDUMP – (COBOL II) formatted dump for ABEND
	■ INTDATELIL|INTDATEANSI – (not COBOL II) Lilian for date basis
	■ LIB|NOLIB – process COPY, BASIS & REPLACE

COBOL (cont.)	■ LIST|NOLIST – compiler listing of generated code
	■ MAP|NOMAP – DATA DIVISION map
	■ NAME|NONAME – link-edit NAME for each object module
	■ NUMBER|NONUMBER – line numbers processed in columns 1 to 6
	■ NUMCLSALT|NONUMCLSPRIM – (not COBOL II) numeric class test
	■ NUMPROCMIG|NONUMPROCMI – invalid sign processing for migrate
	■ NUMPROCPFD|NONUMPROCPFD – invalid sign processing for speed
	■ OBJECT|NOOBJECT – object code output to //SYSLIN
	■ OFFSET|NOOFFSET – condensed PROCEDURE DIVISION list
	■ OPTFULL|NOOPT – (not COBOL II) OPTIMIZE(FULL)
	■ OUTDD|NOOUTDD – DISPLAY ddname instead of //SYSOUT
	■ PGMLONGMIX|NOPGMLONGMIX – (not COBOL II) names are ASIS
	■ PGMLONGUPP|NOPGMLONGUPP – (not COBOL II) not truncated
	■ QUOTE|NOQUOTE – quote mark (") is non-numeric delimiter
	■ RENT|NORENT – reentrant object code
	■ RESIDENT|NORESIDENT – (COBOL II) Library Management to load
	■ RMODEANY|RMODE24 – (not COBOL II) RMODE(ANY)
	■ SEQUENCE|NOSEQUENCE – sequence numbers processed
	■ SIZEMAX|NOSIZEMAX – SIZE(MAX) used in compilation
	■ SOURCE|NOSOURCE – source listing
	■ SSRANGE|NOSSRANGE – check subscripts, indexes, and references
	■ TERMINAL|NOTERMINAL – diagnostic messages to //SYSTERM
	■ TESTBLOCK|NOTESTBLOCK – (not COBOL II) TEST(BLOCK)
	■ TESTPATH|NOTESTPATH – (not COBOL II) TEST(PATH)
	■ TESTSTMT|NOTESTSTMT – (not COBOL II) TEST(STMT)
	■ TRUNCBIN|NOTRUNCBIN – binary field truncation protection
	■ TRUNCOPT|NOTRUNCOPT – binary field truncation for performance
	■ VBREF|NOVBREF – verb cross reference
	■ WORD|NOWORD – alternate reserved word list
	■ XREF|NOXREF – symbol cross reference for names
	■ ZWB|NOZWB – remove sign in compare DISPLAY fields
	NOTE: The following options can affect program execution for these compilers: ADV, AWO, CMPR2, DATA, DYNAM, FASTSRT, FDUMP (COBOL II only), OPTIMIZE, OUTDD, NUMPROC, RENT, RESIDENT (COBOL II only), SSRANGE, TEST, TRUNC, and ZWB.

COBOL (cont.)	**COBOL VS and COBOL V4:**
	▪ COUNT|NOCOUNT – (COBOL VS) verb count summary
	▪ ENDJOB|NOENDJOB – free main storage at ENDJOB
	▪ FLOW|NOFLOW – flow trace output for ABEND
	▪ OBJ370|NOOBJ370 – OBJECT COMPUTER is OBJ370
	▪ OPTIMIZE|NOOPTIMIZE – optimize object code
	▪ READYTRACE|NOREADYTRACE – (COBOL VS) READY TRACE
	▪ RESIDENT|NORESIDENT – Library Management to load
	▪ STATE|NOSTATE – statement number for ABEND
	▪ SYMDMP|NOSYMDMP – formatted dump for an ABEND
	▪ TEST|NOTEST – object code for debug
	NOTE: The following options can affect program execution for these compilers: COUNT (COBOL VS only) FLOW, OBJ370 OPTIMIZE, RESIDENT, STATE, and TEST.
GENERATE	Filter and reformat output from the HISTORY subcommand into a form that can be processed by a program or a REXX exec as described briefly below:
	▪ The ** HIST message contains linkage edit attributes for the module
	▪ The PDS250I message shows all bit settings for COBOL VS and V4
	▪ The PDS251I message TYP field contains ASM, PL1, FOR, RPG, REX, C37, MAP, or ??? with the third character overlaid with D (for DB2), I (for IMS), O (for ONLINE), B (for DB2 and ONLINE), or S (for Secondary if the CSECT name is not the same as the module name)
	▪ The PDS255I message contains all run time option bit settings for COBOL II, COBOL for MVS & VM, or COBOL for OS/390 & VM
	▪ The PDS260I message contains all bit settings for COBOL II, COBOL for MVS & VM or COBOL for OS/390 & VM and the TYP field will be II, MVS or OS with the third character overlaid with D (for DB2), I (for IMS), O (for ONLINE), B (for DB2 and ONLINE), or S (for Secondary if the CSECT name is not the same as the module name)
	▪ Most other HISTORY messages are suppressed
	The format of each message with GENERATE is defined in the PDS250I, PDS251I, PDS255I, and PDS260I message section. The **HIST message separator line is changed as shown below:
	▪ ** HIST COBOLTST Linkage attributes
	▪ Col 9: CSECT name
	▪ Col 23: RENT or blank
	▪ Col 28: REUS or blank
	▪ Col 33: REFR or blank
	▪ Col 38: OVLY or blank
	▪ Col 43: TEST or blank
	▪ Col 48: SCTR (scatter loaded) or blank

	■ Col 53: OL (only loadable) or blank
	■ Col 56: NE (not editable) or blank
	■ Col 59: RANY or R24
	■ Col 64: AANY, A31, A24 or A64
	■ Col 69: RL= followed by the real member name for an alias
EXTERN	Load modules with missing external symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
WKEXTERN	Load modules with missing weak symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
PSEUDOREG	Load modules with PSEUDO register symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
LABELREF	Load modules with ENTRY symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
COMMON	Load modules with COMMON area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
PRIVATE	Load modules with PRIVATE area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
CSECT	Load modules with CSECT symbols are desired. The MODULE keyword may also be specified to search for specific symbols.
LKED(partl)	Specifies that if the linkage editor name matches the partial name, Partl, a line identifying the editor is to be listed. Partl may be entered as a partial linkage editor name with 1 to 10 characters.
NOLKED	The HISTORY information listed should not include linkage editor information. Other types of HISTORY output are not affected.
SYSMOD(Partu)	Specifies that user IDR data which matches the partial name, Partu, is to be listed. Partu may be entered as partial IDR data with 1 to 8 characters. If you also enter MODULE(...), the SYSMOD operand applies only to CSECTs that satisfy the MODULE operand.
NOSYSMOD	Specifies that the HISTORY information listed should not include SYSMOD information. Other types of HISTORY output are not affected.

`TRANS(Partt)`	Specifies that CSECTS identified as having been assembled or compiled by the translator defined by the partial name, Partt, are to be listed. Partt may be entered as a partial translator name with 1 to 10 characters. If you also enter MODULE(...), the TRANS operand applies only to CSECTs that satisfy the MODULE operand.
	If a generic translator name from below is entered instead of Partt, the translator types shown below is used instead.
	■ ASM – IFOX, IEV90, or ASMA90 assembler (Translators 52ASM32B88, 52ASM31686, 566896201, 5734AS1, 569623400, 5741SC103, 360SAS037, 360SAS038, or 5734AS100)
	■ ASMA90 – High-level assembler (Translator 569623400)
	■ ASMH – Assembler H (Translators 566896201 and 5734AS1)
	■ COBOL – COBOL V4, COBOL VS, COBOL II or COBOL 370 (Translators 566895801, 566895807, 566895901, 40CB1, 5740CB103, 5752SC104, 40CB-1, 12345, or 12345-1).
	■ COBOLII – COBOL II (Translators 566895801 or 566895901)
	■ COBOLMVS – COBOL for MVS & VM (Translator 566895807)
	■ COBOLOS – COBOL for OS/390 & VM (Translator 5648A2500)
	■ COBOLVS – COBOL VS (Translators 5740CB103, 40CB1, 40CB-1, 12345, or 12345-1).
	■ COBOLV4 – COBOL V4 (Translators 5752SC104, 12345, or 12345-1).
	■ C370 – C/370 (Translators 5688187, 5688216, or 5688040)
	■ FORTRAN – VS FORTRAN or FORTRAN G or H (Translators 5668-806, 5734-FO2, 5734-FO3, 5734-F02, 5748-FO3, or 5796-PKR)
	■ FORTRANG – FORTRAN G (Translators 5734-FO2 or 5734-F02)
	■ FORTRANH – FORTRAN H (Translator 5734-FO3)
	■ FORTRANV – VS FORTRAN (Translators 5668-806, 5748-FO3, or 5796-PKR)
	■ IFOX – VS assembler (Translators 52ASM32B88, 52ASM31686, 15741SC103, 360SAS037, 360SAS038, or 5734AS100)
	■ PASCAL – VS PASCAL (Translator 566876701)
	■ PLS – PLS-III (Translator C'PLS-III' or CA-PLNK 0)
	■ PLX370 – PL/X 370 (Translator PL/X-370)
	■ PL1 – PL/I (Translators 5734-PL1, 5668-910, or 5688-235)
	■ REXX370 – compiled rexx (Translator 569501301)
	■ RPG – RPG (Translator 5740RG1)
	■ SASC – SAS/C (Translators BLD121988, LC370B, SAS/C, SAS/C/, or SDS080888)
`NOTRANS`	HISTORY information listed should not include TRANSLATOR information. Other types of HISTORY output are not affected.
`USERDATA(Partu)`	User IDR data, which matches the partial name Partu, is to be listed. Partu may be entered as partial IDR data with 1 to 8 characters. If you also enter MODULE(...), the USERDATA operand applies only to CSECTs that satisfy the MODULE operand.

NOUSERDATA	HISTORY information listed should not include USERDATA information. Other types of HISTORY output are not affected.
ZAP(Partz)	Zap IDR data, which matches the partial name Partz, is to be listed. Partz may be entered as partial zap IDR data with 1 to 8 characters. If you also enter MODULE(...), the ZAP operand applies only to CSECTs that satisfy the MODULE operand.
NOZAP	HISTORY information listed should not include ZAP information. Other types of HISTORY output are not affected.
ALL	LKED, TRANS, USERDATA, ZAP is the reporting default. It is often used with one of the negative options as in the following example: HISTORY name NOTRANS ALL
NOSYSTEM	System modules are to be filtered out before reporting MAP information as is done for HISTORY GENERATE. System or compiler routines begin with DFH, DFS, DSN, IBM, IEY, IGY, ILB, ISP, or PLI.
MEMBERS	Displays the names of members that satisfy the HISTORY subcommand without changing the current member group.
MEMLIST	Same as ML. Specifies that any member displayed by the HISTORY subcommand is selected for MEMLIST display. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMOD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.
ML	Same as MEMLIST. Specifies that any member displayed by the HISTORY subcommand is selected for MEMLIST display. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.
NEWML	Same as MEMLIST and ML except that the current MEMLIST is reset.
SUBLIST	Any member displayed by the HISTORY subcommand is selected for inclusion in a new sublist. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMOD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.

Remarks

If any CSECTS in a load member were compiled by a COBOL compiler, several compile-time options are listed for each COBOL CSECT (see messages PDS250I and PDS260I for more information).

If a member is not a load module, any ISPF statistics or any SSI information are displayed for the member and any HISTORY keyword information is ignored.

If DFSMS APAR OW29593 is applied on your system, the binder records hours, minutes and seconds as well as the date in the history IDR record. The StarTool FDM PDS064I message reports this information in HH:MM format if it is present:

```
PDS064I Last link-edited on 1999/06/18 by 5695DF108-BINDER ...
 PDS064I Last link-edited on 1999/06/23 14:04 by 5695DF108-BINDER ...
```

```
----------------------- ISPMODE Session Display ------------- ROW 3000 OF 3010
COMMAND ===>                                                      SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ADC2SUB -----------------------
>----->hi adc2sub

** HISTORY  ADC2SUB
PDS061I AMASPZAP update history by CSECT -
ADC2SUB        88/01/29    E230300
ADC2SUB        87/12/01    E230232
PDS062I User-supplied update history by CSECT -
ADC2LSV5       87/01/28    E230128 -- LOG RETRIEVAL 87/01/27 LEL
ADC2LSV1       86/02/26    ADD PATCH AND COPYRITE 86/02/26 MON
PDS063I ChangeMan module ADC2SUB  package CO10000007 date 1987/03/23 time 10:33
PDS064I Last link-edited on 87/03/23  by LKED 566529508  V01 M00
```

```
---------------------- ISPMODE Session# 1 Log# 1 ------------Row 772 to 791 of 891
 COMMAND ===>                                                     CROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.LOAD,VOL=SER=SER002  MEM=COB* ----------------------------
 >------>hist cob* cobol(rent) trans(cobolii)

** HISTORY  COBOLCAP
PDS060I Translator history by CSECT -
DSN0MGF  85/08/09 566895801-COBOLII   V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF IIS      SSRANG OPT      ZWB (PFD) (OPT) RES RENT       31/C

** HISTORY  COBOLCA2
PDS060I Translator history by CSECT -
DSN0MGF  85/08/09 566895801-COBOLII   V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF IIS      SSRANG OPT      ZWB (PFD) (OPT) RES RENT       31/C
```

```
-----------------------ISPMODE Session# 2 Log -------- ROW 1,000 TO 1,020 OF 1,020
COMMAND ===>                                                      L ===> CSR
- DSN=SYS1.LINKLIB,VOL=SER=SP422P  MEM=IDCAMS -------------------------------
>----->hi idcams

** HISTORY  IDCAMS
PDS062I User-supplied update history by CSECT -
IDCSA03        92/10/15    UY73238
IDCTP06        91/10/01    UY59099
IDCEX02        90/12/19    RSI03481661
PDS063I ChangeMan module IDCAMS   package CO10000017 date 1992/10/15 time 10:33
PDS064I Last link-edited on 92/10/15  by LKED 566528408  V03 M03
```

```
>------>hi : cobol(ssrange,test) ml

** HISTORY  COBMVS
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I COBTEST  MVS TEST SSRANG OPT        ZWB (MIG) (STD) RES RENT        31

** HISTORY  COBOLFFF
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF  MVS TEST SSRANG OPT CMPR2 ZWB (MIG) (BIN) RES RENT DYNAM 31
PDS255I Run-time options:
DEBUG, SSRANGE, STAE, NOAIXBLD, NOSPOUT, NORTEREUS, NOLIBKEEP, NOWSCLEAR, NOMI

** HISTORY  COBOL45
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE_
PDS250I H4810001 VSS                 TEST        RES ENDJOB          COBOL

PDS165I Members are: COBMVS, COBOLFFF, COBOL45

PDS193I This group contains 3 members
```

```
 ------------------- Load MEMLIST (Attributes), Session# 1 --- Row 1 to 3
 COMMAND ===>                                          SCROLL ==> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.LOAD,VOL=SER=SER002  MEM=(COBMVS ------------------------
 CMD  NAME     DATA/MSG ALIASOF LEN/LKED  -- ATTRIBUTES  - APF MODE   MAIN
      COBMVS                    1998/04/23                     AANY
      COBOLFFF                  1991/06/25 RENT REUS
      COBOL45                   1998/05/05
      TESTAM64                  2003/01/24                     A64
```

```
--------------------- ISPMODE Session# 2 Log ----------ROW 1,000 TO 1,023 OF 1,023
COMMAND ===>                                            ROLL ===> CSR
- DSN=SER07.LIB.LOAD,VOL=SER=SER007  MEM=(COBOL6 ---------------------------
>----->hi (cobola,cobolran)

** HISTORY  COBOLA
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE
PDS250I CZARVSY VS2      FLOW        TEST        RES ENDJOB SYMD 370 OPTIMIZE
PDS062I User-supplied update history by CSECT -
DFSLI000    83/06/16    RSI31640356
PDS063I ChangeMan module COBOLA   package CO10000027 date 1989/12/01 time 10:33
PDS064I Last link-edited on 89/12/01  by LKED 566528408  V02 M03

** HISTORY  COBOLRAN
PDS260I CSECT___VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF IID       SSRANG OPT        ZWB NOPFD (STD) RES RENT        31/CA
PDS260I DSN0MGZ IID       SSRANG OPT        ZWB (PFD) (BIN) RES RENT        31/CA
PDS255I Run-time options:
DEBUG, SSRANGE, STAE, NOAIXBLD, NOSPOUT, NORTEREUS, NOLIBKEEP, NOWSCLEAR, NMIX
PDS062I User-supplied update history by CSECT -
IGZEBST     85/05/06    UP52062
ISPLINK     84/11/26    RSI43240150
```

```
>------>hi (cobol6,dsx0mgf,cobos390) nosyst

** HISTORY  COBOL6
PDS060I Translator history by CSECT -
HM01A    1985/01/29 40CB1-COBOLVS      V20 M00
DATECALC 1983/06/20 5734AS100-IFOX00   V05 M01
UABEND   1983/06/20 5734AS100-IFOX00   V05 M01
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE_
PDS250I HM01A    VSS                             ENDJOB    370 CAPEX
PDS061I AMASPZAP update history by CSECT -
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS063I ChangeMan module COBOL6   package CO10000027 date 1989/12/01 time 10:33
PDS064I Last link-edited on 1989/12/01 by 566528408-DFPLKED    V02 M03

** HISTORY  DSX0MGF
PDS060I Translator history by CSECT -
DSX0MGF  1985/08/09 566895801-COBOLMVS  V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSX0MGF  IID      SSRANG OPT      ZWB (PFD) (OPT) RES RENT       31
PDS061I AMASPZAP update history by CSECT -
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS063I ChangeMan module DSXOMGF   package CO10000127 date 1991/06/25 time 10:33
PDS064I Last link-edited on 1991/06/25 by 566528408-DFPLKED    V03 M01

** HISTORY  COBOS390
PDS060I Translator history by CSECT -
COBTEST  1998/04/23 5648A2500-COBOLOS   V12 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I COBTEST  OSS                     ZWB NOPFD (STD) RES           31
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS063I ChangeMan module COBOS390 package CO10000327 date 1998/04/23 time 10:33
PDS064I Last link-edited on 1998/04/23 by 5695DF108-BINDER    V01 M01
```

```
>------>hi (cobol6,dsx0mgf,cobos390) gen

** HIST COBOL6                                          R24  A64
PDS250I HM01A    VSS  NNNNNNNNNNNNNNNNYNYYYNNNNN
PDS251I DATECALC ASS
PDS251I UABEND   ASS

** HIST DSX0MGF       RENT REUS                         RANY A31  RL=BOBZIM
PDS260I DSX0MGF  IID  NYYNYNNNNNYNYYYNYYYNYYYYNNNNYYNNNNNNNNNN
PDS255I Run-time options: NNNNNNNNNNNYYYNNNNNNNNNNNNNNNNNNNNNNNNNN

** HIST COBOS390                                        R24  AANY
PDS260I COBTEST  OSS  YNYNNNNNNNNNYNNNNNYYYYNNNYNNNYNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

IDCAMS Subcommand

The IDCAMS subcommand invokes IDCAMS passing it a command and operands and
displays the output from the command.

Example

```
IDCAMS   listcat   entry(userid.vsam.ix.cluster)   all
```

Syntax

```
IDCAMS   command   operands
```

Aliases

```
ID, IDC, IDCA, IDCAM, IDCAMS
```

Defaults

Required

IDCAMS command and operands; if no IDCAMS command is entered in ISPF mode, an IDCAMS prompting sequence is initiated.

Operands

command	Specifies the IDCAMS command: ALLOCATE, ALTER, BLDINDEX, CNVTCAT, DEFINE, DELETE, EXAMINE, EXPORT, IMPORT, LISTCAT, PRINT, REPRO, or VERIFY. IDCAMS is invoked as an APF-authorized processor through the TSO/E IKJEFTSR interface. Operations on a catalog that require authorization (DEFINE with RECATALOG, DEFINE with RECOVERY, EXPORT, IMPORT, PRINT, REPRO, or VERIFY) can be performed.
operands	Any additional operands for the command. For example, DEFINE needs an initial operand of ALIAS, AIX, CLUSTER, GDG, NONVSAM, PAGESPACE, PATH, or USERCATALOG followed by additional operands to complete the DEFINE command.

Remarks

If you invoke the IDCAMS subcommand with no operands in an ISPF environment, you are prompted for which of several IDCAMS commands you want to perform. Previous operands entered for each command are remembered across ISPF sessions.

An IDCAMS subcommand issued from StarTool FDM is limited to a maximum of 245 characters of data. This restriction may be a problem if you are entering IDCAMS DEFINE commands directly. The IDCAMS DEFINE prompts use an alternative interface that allows up to 800 characters.

If you use the IDCAMS prompting interface, normal TSO conventions are used for data set names but catalog names must be fully qualified. When IDCAMS is invoked with a command and operands, all data set and catalog names must be fully qualified but you can optionally use quotation marks around data set names.

DEFINE is a short-hand command for entering IDCAMS with no operands and selecting the DEFINE option.

The following DEFINE KSDS panel can be reached by:

IDCAMS;DEFINE;KSDS

or

DEFINE;KSDS

```
----------------------- IDCAMS DEFINE for a KSDS  -----------------------------
OPTION  ===>

 KSDS cluster name ==> VSAM.KSDS
 Volume name       ==> SER006    (optional if SMS managed or modelled)
 Model data set    ==>
 Average record    ==> 50
 Maximum record    ==> 500
 Span records      ==> NO        (YES or NO)
 Expiration date   ==>           (Julian -- yyyyddd)
 Retention time    ==>           (number of days, this overrides expiration)
 DATA space type   ==> CYLINDERS (CYLINDERS/KILOBYTES/MEGABYTES/RECORDS/TRKS)
      primary      ==> 1         (in above units)
      secondary    ==> 1         (in above units)
 INDEX space type  ==> TRACKS    (CYLINDERS/KILOBYTES/MEGABYTES/RECORDS/TRKS)
       primary     ==> 1         (in above units)
       secondary   ==> 1         (in above units)
 Key length        ==> 26        (1 to 255)
 Key offset        ==> 0         (0 to end of record)
 SMS Data class    ==>
 SMS MGMT class    ==>
 SMS Storage class ==>
 More parameters?  ==> YES       (YES or NO)
```

```
---------------------- IDCAMS DEFINE for a KSDS (part 2) --------------------
OPTION  ===>

 DATA component     ==> 'SER07.VSAM.KSDS.DATA'
 INDEX component    ==> 'SER07.VSAM.KSDS.INDEX'
 Erase on delete    ==> NO       (YES or NO)
 Recatalog data set==> NO        (YES or NO)
 Replicate index    ==> NO       (YES or NO)
 Reusable cluster   ==> NO       (YES or NO)
 Imbed index        ==> NO       (YES or NO)
 Speed load         ==> YES      (YES or NO)
 Write check        ==> NO       (YES or NO)
 Owner              ==> SER07    (owner-id of cluster)
 DATA CI size       ==> 4096     (multiple of 512 to 8K; multiple of 2K to 32K)
 INDEX CI size      ==> 4096     (multiple of 512 to 8K; multiple of 2K to 32K)
 Buffer space       ==>          (minimum: two DATA CI's plus one INDEX CI)
 CI percent free    ==> 10       (0 to 100)
 CA percent free    ==> 15       (0 to 100)
 Crossregion share ==> 1         (1, 2, 3 or 4)
 Crosssystem share ==> 3         (3 or 4)
 Catalog name       ==>
```

IF Subcommand

The IF subcommand searches for members that meet defined conditions. The action to be taken is specified by the THEN and ELSE keywords. You can omit the THEN or ELSE keyword. If you omit both THEN and ELSE, a default of THEN(ATTRIB) is assumed.

If all conditions are met for a given member, any THEN action is taken for that member; otherwise, any ELSE action is taken.

Example

```
IF   mema:memb  changed(93/11/24:93/11/30)  then(edit)
```

Syntax

```
IF memgroup
    [SINCE|BEFORE                                      ]
    [TODAY|YESTERDAY|WEEK|CURRENT|BIWEEK|
            MONTH|QUARTER|HALFYEAR|YEAR|BIYEAR|
            LAST(numdays)|DATE(yyyy/mm/dd)             ]
    [CHANGED(yyyy/mm/dd:yyyy/mm/dd)                    ]
    [CREATED(yyyy/mm/dd:yyyy/mm/dd)                    ]
    [ABOVE(Count1)                                     ]
    [ALIAS|NOALIAS                                     ]
    [AMODE24|AMODE31|AMODE64|AMODEANY|NOAMODE24|NOMODE31
            NOAMODE64|NOAMODEANY                       ] (load only)
 *  [APFERR|NOAPFERR                                   ] (load only)
 *  [APPARENTALIAS|NOAPPARENTALIAS                     ]
    [AUTH|NOAUTH                                       ] (load only)
    [BELOW(Count2)                                     ]
 *  [BLOCKERR|MAXBLK(size)|NOBLOCKERR                  ]
    [DC|NODC                                           ] (load only)
    [EDIT|NOEDIT                                       ] (load only)
    [EXEC|NOEXEC                                       ] (load only)
 *  [EXTERN|WKEXTERN|PSEUDOREG|LABELREF|COMMON|PRIVATE|CSECT] (load)
    [FLEVEL|NOFLEVEL                                   ] (load only)
    [HASALIAS|NOHASALIAS                               ]
    [ID(Puid)|NOID|NOTID(Puid)                         ]
 *  [IDRFULL|NOIDRFULL                                 ] (load only)
 *  [IOERR|NOIOERR                                     ]
    [LKED(Partl)                                       ] (load only)
 *  [LKEDERR|NOLKEDERR                                 ] (load only)
 *  [LOADERR|NOLOADERR                                 ] (load only)
    [LOADONLY|NOLOADONLY                               ] (load only)
 *  [LRECLERR|NOLRECLERR                               ]
    [MODULE({* | Fullm | Partm*)}                      ] (load only)
 *  [NAMEERR|NONAMEERR                                 ]
    [NULL|NONULL                                       ]
    [ORPHAN|NOORPHAN                                   ]
    [OVERLAY|NOOVERLAY                                 ] (load only)
   *[PACKED|NOPACKED                                   ]
    [PAGE|NOPAGE                                       ] (load only)
    [REFR|NOREFR                                       ] (load only)
    [RENT|NORENT                                       ] (load only)
    [REUS|NOREUS                                       ] (load only)
 *  [RLDERR|NORLDERR                                   ] (load only)
 *  [RLDZERO|NORLDZERO                                 ] (load only)
    [RMODE24|RMODEANY|NORMODE24|NORMODEANY             ] (load only)
    [SCTR|NOSCTR                                       ] (load only)
 *  [SPFEDIT|NOSPFEDIT                                 ]
    [SSI(hexdata)|SSI|NOSSI|PARTSSI(hexdata)           ]
    [SYSMOD(Partu) | USERDATA(Partu)                   ] (load only)

    [TEST|NOTEST                                       ] (load only)
    [TRANS(Partt)                                      ] (load only)
```

```
          [TTR(Lttr:Httr)                                    ]
          [USERDATA(Partu) | SYSMOD(Partu)                   ] (load only)
          [USERID(Puid)|NOUSERID|NOTUSERID(Puid)             ]
          [VSLKED|NOVSLKED                                    ] (load only)
          [ZAP(Partz)                                        ] (load only)
         [THEN(ATTRIB | BROWSE   | DIRENTRY | DELETE   | EDIT     |
                       EXCLUDE | FIND     | FSE      | HISTORY | LIST    |
                       MAP     | MEMBERS  | MEMLIST  | NEWML   | PRINT   |
                       REVIEW  | SUBMIT   | SUBLIST  | TSOEDIT | VERIFY  |
                       VIEW                                    ]
         [ELSE(ATTRIB | BROWSE   | DIRENTRY | DELETE   | EDIT     |
                       EXCLUDE | FIND     | FSE      | HISTORY | LIST    |
                       MAP     | MEMBERS  | MEMLIST  | NEWML   | PRINT   |
                       REVIEW  | SUBMIT   | SUBLIST  | TSOEDIT | VERIFY  |
                       VIEW                                    ]
```

```
* Note: the * marked tests are unique to IF as compared to ML or EXCLUDE
```

Aliases

I, IF

Defaults

memgroup, SINCE, THEN(ATTRIB) if neither THEN nor ELSE is entered

Required

Operands

memgroup	Identifies the members whose attributes are to be examined. Default member names, member lists, member name ranges, and member name patterns are allowed; see Appendix A, "Formatting Member Names".
SINCE	Uses dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
FROM	Uses dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
BEFORE	Uses dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
TO	Uses dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
TODAY	A date, equivalent to LAST(0).
YESTERDAY	A date, equivalent to LAST(1).
WEEK	A date, equivalent to LAST(7).
CURRENT	A date, equivalent to LAST(10).
BIWEEK	A date, equivalent to LAST(14).
MONTH	A date, equivalent to LAST(30).
QUARTER	A date, equivalent to LAST(120).
HALFYEAR	A date, equivalent to LAST(183).
YEAR	A date, equivalent to LAST(365).
BIYEAR	A date, equivalent to LAST(730).
LAST(numdays)	A date indicating the number of days before today.
DATE(cdate)	A date indicating the actual date. It can be entered in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).
CHANGED(date:	Searches for members modified by an ISPF editor or a ZAP program between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like: `CHANGED(yyyy/mm/dd:yyyy/mm/dd)` Only members with ISPF statistics or members created by a linkage editor are supported.

CREATED(date:	Searches for members created with ISPF statistics or by the linkage editor between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like: `CREATED(yyyy/mm/dd:yyyy/mm/dd)`
ABOVE(Count1)	Selects members with more than the number of lines (amount of storage for load modules) defined by the number, Count1.
ALIAS	Selects alias members.
NOALIAS	Selects main members.
AMODE24	Selects modules with addressing mode 24.
NOAMODE24	Selects modules with addressing mode 31, 64 or ANY.
AMODE31	Selects modules with addressing mode 31.
NOAMODE31	Selects modules with addressing mode 24, 64 or ANY.
AMODE64	Selects modules with addressing mode 64.
NOAMODE64	Selects modules with addressing mode 24, 31 or ANY.
AMODEANY	Selects modules with addressing mode ANY.
NOAMODEANY	Selects modules with addressing mode 24, 31, or 64.
APFERR	Selects modules linked by the system linkage editor with invalid APF length indicators.
NOAPFERR	Selects modules linked by the system linkage editor with valid APF length indicators or modules linked with previous linkage editors.
APPARENTALIAS	Selects main members whose start addresses match the start address of another main member.
NOAPPARENTALIAS	Selects alias members or main members whose start addresses do not match the start address of another main member.
AUTH	Selects APF-authorized modules.
NOAUTH	Selects non-authorized modules.
BELOW(Count2)	Selects members with less than the number of lines (amount of storage for load modules) defined by the number, Count2.
BLOCKERR	Selects members with block sizes exceeding the DCB BLKSIZE.
MAXBLK(sizeb)	Selects members with block sizes exceeding sizeb.
NOBLOCKERR	Selects members without block size errors.
DC	Selects downward-compatible modules.
NODC	Selects no downward-compatible modules.
EDIT	Selects modules that can be reprocessed by the linkage editor.
NOEDIT	Selects modules that cannot be reprocessed by the linkage editor.
EXEC	Selects executable modules.

NOEXEC	Selects non-executable modules.
EXTERN	Selects load modules with missing external symbols. The MODULE keyword may also be specified to search for specific missing symbols.
WKEXTERN	Selects load modules missing weak external symbols. The MODULE keyword may also be specified to search for specific missing symbols.
PSEUDOREG	Selects load modules with PSEUDO register symbols. The MODULE keyword may also be specified to search for specific symbols.
LABELREF	Selects load modules with ENTRY symbols. The MODULE keyword may also be specified to search for specific symbols.
COMMON	Selects load modules with COMMON area symbols. The MODULE keyword may also be specified to search for specific symbols.
PRIVATE	Selects load modules with PRIVATE area symbols. The MODULE keyword may also be specified to search for specific symbols.
CSECT	Selects load modules with CSECT symbols. The MODULE keyword may also be specified to search for specific symbols.
FLEVEL	Selects modules processed by the F-level linkage editor.
NOFLEVEL	Selects modules not processed by the F-level linkage editor.
HASALIAS	Selects main members that have one or more aliases.
NOHASALIAS	Selects alias members or main members without any aliases.
ID(Puid)	Selects members with ISPF statistics and user IDs matching the partial name Puid. Puid can be entered as a partial TSO user ID with 1 to 7 characters. ID is an alias of USERID.
NOID	Selects members without ISPF statistics. NOID is an alias of NOUSERID.
NOTID(Puid)	Selects members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name Puid. Puid can be entered as a partial TSO user ID with 1 to 7 characters. NOTID is an alias of NOTUSERID.
IDRFULL	Selects members that cannot contain more IMASPZAP IDR records.
NOIDRFULL	Selects members that contain more IMASPZAP IDR records.
IOERR	Selects members with I/O errors.
NOIOERR	Selects members without I/O errors.
LKED(Partl)	Selects modules identified as linked by the linkage editor defined by the partial name Partl. Partl can be entered as a partial linkage editor IDR name with 1 to 10 characters.

LKEDERR	Selects modules with any of the following illegal linkage edit attribute combinations: ■ Modules with RENT and NOREUS attributes ■ Modules with RMODEANY and AMODE24 or AMODEANY attributes ■ Modules with OVLY and RENT, REUS, REFR, SCTR, RMODEANY, AMODE31, AMODE64, or AMODEANY attributes ■ Modules with TEST and NOEDIT attributes ■ Modules with REUS and SCTR attributes
NOLKEDERR	Selects modules with none of the above illegal linkage edit attribute combinations.
LOADERR	Selects modules that ABEND when they are LOADED.
NOLOADERR	Selects modules that can be LOADED without errors.
LOADONLY	Selects modules marked for LOAD ONLY.
NOLOADONLY	Selects modules not marked for LOAD ONLY.
LRECLERR	Selects members with any of the following LRECL errors: ■ BLKSIZE divided by input LRECL not integral (RECFM=F) ■ Input LRECL exceeding maximum DCB LRECL (RECFM=V) ■ Input LRECL less than 4 bytes (RECFM=V)
NOLRECLERR	Selects members with none of the above LRECL errors.
MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT or ENTRY names for attribute searches. If you enter this parameter with any TRANS, SYSMOD/USERDATA, or ZAP keywords, only those CSECTs that satisfy the MODULE keyword are checked for the other keyword conditions. The MODULE operand has several valid forms: ■ MODULE(*) – uses the previous name entered on any MODULE keyword ■ MODULE(Fullm) – considers only modules containing a CSECT or ENTRY named FULLM ■ MODULE(Partm*) – considers only modules containing a CSECT or ENTRY named PARTM...
NAMEERR	Selects members whose member names are not upper-case alphanumeric or whose first character is numeric.
NONAMEERR	Selects members whose member names are upper-case alphanumeric and whose first character is not numeric.
NULL	Selects members with no data.
NONULL	Selects members with some data.
ORPHAN	Selects alias members that have no associated main member.
NOORPHAN	Selects main members or alias members that have an associated main member.
OVERLAY	Selects overlay modules.
NOOVERLAY	Selects non-overlay modules.

PACKED	Specifies that ISPF packed source members are desired.
NOPACKED	Specifies that ISPF packed source members are not desired.
PAGE	Selects modules marked for loading on a page boundary.
NOPAGE	Selects modules not marked for loading on a page boundary.
REFR	Selects refreshable modules.
NOREFR	Selects non-refreshable modules.
RENT	Selects reentrant modules.
NORENT	Selects non-reentrant modules.
REUS	Selects reusable modules.
NOREUS	Selects non-reusable modules.
RLDERR	Selects modules linked since OS/VS whose RLD/CONTROL count from the directory does not match the first RLD entry.
NORLDERR	Selects modules whose RLD/CONTROL count from the directory match the first RLD entry or modules linked before OS/VS.
RLDZERO	Selects modules whose RLD/CONTROL count is zero.
NORLDZERO	Selects modules whose RLD/CONTROL count is not zero.
RMODE24	Selects modules with residence mode 24.
NORMODE24	Selects modules with residence mode ANY.
RMODEANY	Selects modules with residence mode ANY.
NORMODEANY	Selects modules with residence mode 24.
SCTR	Selects scatter-loaded modules.
NOSCTR	Selects no scatter-loaded modules.
SPFEDIT	Selects members that are currently in use by an ISPF EDIT session.
NOSPFEDIT	Selects members that are not currently in use by an ISPF EDIT session.
SSI(hexdata)	Selects members with matching SSI data. This is implemented as a generic match of SSI characters on the first part of the SSI field for the number of characters entered. For example, SSI(698) would match actual SSI fields of 698BADEF or 69800000 but it would not match 06980000.
SSI	Selects members with SSI data.
NOSSI	Selects members without SSI data.
PARTSSI(hx)	Selects members with matching SSI data. This is implemented as a pattern check, matching SSI digits as a string anywhere in the SSI field of the member. For example, PARTSSI(698) would match actual SSI fields of 698BADEF or 00006980 but it would not match 80000069.

SYSMOD(Partu)	Selects modules with user IDR data that matches the partial name Partu. Partu can be entered as partial IDR data with 1 to 8 characters. If you also enter MODULE(...), the SYSMOD operand applies only to CSECTs whose names satisfy the MODULE argument.
TEST	Selects modules with the TEST attribute.
NOTEST	Selects modules without the TEST attribute.
TRANS(Partt)	Selects modules with CSECTS identified as having been assembled or compiled by the translator defined by the partial name Partt. Partt can be entered as a partial translator IDR name with 1 to 10 characters. If you also enter MODULE(...), the TRANS operand applies only to CSECTs whose names satisfy the MODULE argument.
TTR(Lttr:Httr)	Selects members whose start address is in the specified TTR range. Lttr defaults to 0 and can be entered as a hexadecimal TTR value from 0 through FFFFFF. Httr defaults to FFFFFF and can be entered as a hexadecimal TTR value from 0 through FFFFFF.
USERDATA(Partu)	Selects modules with user IDR data that matches the partial name Partu. Partu can be entered as partial IDR data with 1 to 8 characters. If you also enter MODULE(...), the USERDATA operand applies only to CSECTs whose names satisfy the MODULE argument.
USERID(Puid)	Selects members with ISPF statistics and user IDs matching the partial name Puid. Puid can be entered as a partial TSO user ID with 1 to 7 characters. USERID is an alias of ID.
NOUSERID	Selects members without ISPF statistics. NOUSERID is an alias of NOID.
NOTUSERID(Puid)	Selects members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name Puid. Puid can be entered as a partial TSO user ID with 1 to 7 characters. NOTUSERID is an alias of NOTID.
VSLKED	Selects modules linked by the system linkage editor.
NOVSLKED	Selects modules not linked by the system linkage editor.
ZAP(Partz)	Selects modules with zap IDR data that matches the partial name Partz. Partz can be entered as partial zap IDR data with 1 to 8 characters. If you also enter MODULE(...), the ZAP operand applies only to CSECTs whose names satisfy the MODULE argument.

THEN(action)	Executes a subcommand if all conditions are met for a member. The subcommand can be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FIND, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY, or VIEW. NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET. If neither a THEN nor an ELSE operand is entered, THEN(ATTRIB) is the default condition.
ELSE(action)	Executes a subcommand if any condition is *not* met for a member. The subcommand can be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FIND, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY, or VIEW. NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET. If neither a THEN nor an ELSE operand is entered, THEN(ATTRIB) is the default condition.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
---------------------- ISPMODE Session Display -------------- ROW 3011 OF 3020
COMMAND ===>                                                    LL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=:  ------------------------------
>----->if : rmodeany then(sublist).
>----->members
PDS165I Members are: ASID, ASIDZN, PDSPGM, WHAT, WHATO

PDS193I This group contains 5 members

>----->if : rmodeany then(attr)
PDS232I NAME     ALIASOF   CREATED   SIZE SSI      ATTRIBUTES
PDS232I ASID               87/06/15  3120          RANY, A31
PDS232I ASIDZN   ?UNKNOWN 87/06/15  3120 CB123021 RANY, A31
PDS232I PDSPGM             89/06/10  436K          RANY, A31, RENT, REUS
PDS232I WHAT     PDSPGM    89/06/10  436K          RANY, A31, RENT, REUS
PDS232I WHATO              89/02/17  388K          RANY, A31, RENT, REUS

PDS119I 3 members RMODEANY; size is 827K
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
---------------------- ISPMODE Session# 2 Log# 1 -------  ROW 735 TO 747 OF 747
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=:  --------------------------------
>----->if : last(15)
PDS230I ISPF Stats: VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
PDS230I  A          01.00  93/04/08  93/04/08 13:21    5     5     0   SER07
PDS230I  ABLK        01.01  93/04/09  93/04/09 14:37    6     9     0   SER07
PDS230I  ADOC        01.01  93/04/15  93/04/15 10:17  156   192     0   SER07
PDS230I  LOADMARK    01.02  93/02/26  93/04/05  6:43   50    48     0   SER07
PDS230I  SAMPSECR    01.26  90/03/27  93/04/16  5:57  197   220   140   SER07
PDS230I  TRIAL601    01.00  93/04/14  93/04/14 15:16  123   123     0   SER07
PDS117I 6 members counted; cumulative size from statistics is 537 records

PDS147I 292 members searched
PDS148I 6 members found
```

ISPF Subcommand

The ISPF subcommand invokes the ISPF primary panel with any option.

Example

```
ISPF    3.3
```

Syntax

```
ISPF    [option]
```

Aliases

```
IS, ISP, ISPF
```

Defaults

Required

Operands

option	Specifies an initial ISPF option (such as 3.3) that allows you to skip intermediate panel displays.

Remarks

The result of using the ISPF subcommand is a temporary suspension of your StarTool FDM session and entry to the specified ISPF function. After you terminate ISPF by repeated ENDs, your StarTool FDM session is activated again. This is a stacked ISPF session.

ISPMODE Function

The ISPMODE subcommand switches StarTool FDM to a full screen session display mode. Subcommands entered and the outputs from internal StarTool FDM subcommands are maintained in a log.

Example

```
ISPMODE
```

Syntax

```
ISPMODE
```

Aliases

```
ISPM, ISPMO, ISPMOD, ISPMODE
```

Operands

No operands are supported for the ISPMODE subcommand.

Remarks

The ISPMODE log displays in response to a ISPMODE subcommand issued in line mode. You then enter ISPMODE (an ISPF dialog mode). After you are in dialog mode, use the LOG command to select any one of nine different logs for your session display.

When ISPMODE is active, StarTool FDM operates as a normal ISPF dialog. Enter the HELP command while in ISPMODE to get ISPF tutorial information or enter the INDEX command to go to the StarTool FDM tutorial index panel directly.

Suspend ISPMODE with the SUSPEND command. SUSPEND places your session into line mode (for example, you can go into line mode to SEND or RECEIVE data using IND$FILE). To enter ISPMODE again, type an ISPMODE, ISPXEQ, or MEMLIST subcommand.

ISPXEQ Function

Use the ISPXEQ subcommand to enter StarTool FDM in LISTA or LISTV dialog mode instead of the usual ISPMODE or MEMLIST mode.

ISPXEQS is a variation of ISPXEQ that invokes a "short" variation. StarTool FDM terminates at the first END command.

Example

```
STARTOOL  LIB.CNTL  ISPXEQ  LISTA  OPEN
```

Syntax

```
ISPXEQ | ISPXEQS   {command}
```

Aliases

```
ISPX, ISPXE, ISPXEQ, ISPXEQS
```

Defaults

Required

command

Operands

command	Specifies a ISPMODE command to be executed after StarTool FDM initialization.

Remarks

Use the ISPXEQ subcommand in building ISPF command table entries for StarTool FDM dialog functions. For example, add the following command table entry for an ISPF command called LA:

```
SELECT CMD(STARTOOL 'SYS1.HELP' ISPXEQ LISTA &ZPARM)
```

Commands — L to N

LIST Subcommand

The LIST subcommand displays data from a member or data set.

Example

```
LIST    mema:memb
```

Syntax

```
LIST    memgroup [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP         ]
                 [OFFSET(Hx) MODULE({* | Fullm | Partm*)}        ]
                 [MAXMEMBERS(num)                                ]
                 [FORMAT(from:to,from:to, ...)                   ]
                 [SKIPREC(n) MAXIN(n) MAXOUT(n)                  ]
                 [SKIPCOL(n) MAXLEN(n)                           ]
               * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)  ]
               * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)        ]

  *NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases

```
LI, LIS, LIST
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

memgroup, NUM or previous LIST|FIND|REPLACE format

Required

Operands

memgroup	Identifies the members to be displayed.
	Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
NUM	Examines the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record. If the field is not numeric, switches formatting to NONUM mode for the remainder of the member.
	For ISPF-saved members, the high-order 6 digits of the line number field is formatted; otherwise, the low-order 6 digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from a logical record.
SNUM	Discards the line number field (the last 8 bytes for a fixed-format record or the first 8 otherwise). Displays up to 256 bytes from a logical record.
NONUM	Displays up to 256 bytes from a logical record (without regard to line numbers).
LBLOCK	Formats logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks.
	For load modules, only CSECT data is displayed. If you enter a name in a MODULE operand, only those CSECT or ENTRY names that satisfy the condition display. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks.
	For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + will be placed after the key); otherwise, it is a relative byte address.
LDUMP	Logical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.
	For load modules, only CSECT data displays. If you enter a name in a MODULE operand, only those CSECT or ENTRY names that satisfy the condition display. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks.
	Also, for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + will be placed after the key); otherwise, it is a relative byte address.

BLOCK	Physical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + will be placed after the key); otherwise, it is a relative byte address. If control interval processing is being performed, the first field is always the control interval relative byte address.
DUMP	Physical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer 8 eight characters, a + will be placed after the key); otherwise, it is a relative byte address. If control interval processing is being performed, the first field is always the control interval relative byte address.
OFFSET(Hx)	Specifies a 1- to 6-digit hexadecimal module offset at which the display is to begin. This operand is only defined for load modules and only with the LDUMP or LBLOCK output formats. If both MODULE and OFFSET keywords are entered, the offset applies to each module selected.
MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT or ENTRY names to be displayed. This operand is only defined for load modules and only with the LDUMP or LBLOCK output formats. If both MODULE and OFFSET keywords are entered, the offset applies to each module selected. The MODULE operand has several valid forms: ■ MODULE(*) – uses the previous name entered on any MODULE keyword ■ MODULE(Fullm) – displays only a CSECT or ENTRY named FULLM ■ MODULE(Partm*) – displays only a CSECT or ENTRY named PARTM...
MAXMEMBERS(n)	*n* is coded as 0 through 9999999 to specify the maximum number of members to be output by LIST.
FORMAT (from:to,from:to, ...)	The record columns to display in the output when displaying LIST data. FORMAT is only supported with NUM, SNUM or NONUM data. For example, FORMAT(21:30,0,1:10) specifies that the output should be formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.
SKIPREC(n)	Ignores *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.
MAXIN(n)	Inputs up to *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.

MAXOUT(n)	Displays up to *n* (coded as 0 through 9999999) output lines for a member.
SKIPCOL(n)	Ignores *n* (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats). For NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.
MAXLEN(n)	Displays up to *n* (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).
FROMKEY(ky)	For VSAM data sets only, ky is coded as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'. Access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. It can only be specified for an alternate index or a key-sequenced data set.
FROMADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Can be specified for any VSAM data set component if control interval processing is being used ■ Cannot be specified if the data set is being accessed through a path. ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. It can only be specified for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is coded as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. It can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Can be specified for any VSAM data set component if control interval processing is being used ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. This parameter can be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks

Unprintable characters for 3270-type devices are translated to periods before they display.

LIST formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, and DUMP. For load modules, the NUM, SNUM, and NONUM formats are equivalent to the LDUMP format. For VSAM data sets, the NUM, SNUM, and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time a format operand is entered on a LIST, FIND, or REPLACE subcommand, the value entered is used as the output format for subsequent LIST, FIND, and REPLACE subcommands. Also, note the following:

1 Formats NUM, SNUM, and NONUM limit the display length for logical records to 256 characters.

2 Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is being performed). The other formats apply to logical records. For load modules, LBLOCK and LDUMP formats display only CSECT data.

3 For VSAM DATA or INDEX components, the LIST, FIND, and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

Control interval access could be useful if a VSAM data set has logical errors. REPLACE can repair the error; however, since only the component is opened for update, the next access of the data set through the related cluster gets warning errors due to the differing time stamps.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------- ISPMODE Session Display ----------------  ROW 7 OF 682
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAPHELP ------------------------
>----->list zaphelp off(40c0)
PDS141I AT 003FF0  CSECT CSOUT     LENGTH 00000428
  0040C0 00D0  45F0C0E4 000040CC  00000000 C9D2D1C4  *.0U.. .....IKJD*
  0040D0 00E0  C1C9D940 0A0686FF  C03AD207 A090D1A0  *AIR ..f..K...J.*
  0040E0 00F0  D703A010 A0104100  A0305000 A008928F  *P.........&...k.*
  0040F0 0100  A008D25F A030C3A8  D207A058 A0904110  *..Kª..CyK.......*
  004100 0110  A0080A13 9110A060  4710C1CA 41F00004  *....j..-..A..0..*
  004110 0120  47F0C03A 4110A030  58F01030 45E0F008  *.0......0....*
  004120 0130  47F0C1CA 4800A012  95F1A0BF 4780C1D0  *.0A.....n1....A*
  004130 0140  95F0A0BF 4780C162  954EA0BF 4780C166  *n0....A.n+....A.*
  004140 0150  9560A0BF 4780C160  9240A0BF 47F0C164  *n-....A-k ...0A.*
  004150 0160  06000600 06001200  47D0C1D0 4000A012  *.........A ...*
  004160 0170  4110A030 4100A0BF  58F01030 05EF9120  *.........0....j.*
  004170 0180  A0004780 C1B6D201  A0BEC422 4110A13B  *....A.K...D... .*
  004180 0190  95401000 4770C19C  4610C190 D2031001  *n ....A...A.K...*
  004190 01A0  C41C4100 A0BB1B10  47D0C1B6 4010A0BC  *D........A. ...*
  0041A0 01B0  4110A0BC 0A239104  A0004780 C1CA94FB  *......j.....A.m.*
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  EATURES
----------------- ISPMODE Session# 2 Log ---------- ROW 1,000 TO 1,017 F 1,176
COMMAND ===>                                              SCROLL ==> CSR
- DSN=SER07.LIB.CLISTV,VOL=SER=SER006  MEM=CLI* ----------------------------
>----->list cli*

** LIST    CLIST
 /* THE FOLLOWING CLIST WILL ESTABLISH AN ALTLIB FOR CLIST TESTING   */
 /*                                                                  */
 PROC 1 DSNAME
 ALTLIB  ACT    APP(CLIST)      DSN(&DSNAME)
 WRITE &DSNAME IS A NEW CLIST LIBRARY FOR TESTING
PDS142I 5 lines in this member

** LIST    CLISTDSP
000100  PROC 2 DSN MEM VOL(VOLSER)
000200  ISPEXEC DISPLAY PANEL(&MEM.)
PDS142I 2 lines in this member

** LIST    CLISTSOE
PROC 0
CONTROL MAIN
```

```
------------------------ ISPMODE Session# 1 Log# Row 6,031 to 6,050 of 9,034
     COMMAND ===>                                        SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.SDSF.LIST,VOL=SER=SER007  -------------------------------------
 >------>list format(9:22,0,41:120)
9501 7AB7               CLI    #FIXORDR,1           ORDER KEYWORD?
4770 835C               BNE    DIRO920              NO, SKIP
9102 7864               TM     DSORG,DS1DSGPO       PARTITIONED?
47E0 835C               BNO    DIRO920              NO, SKIP
9200 7AB7               MVI    #FIXORDR,0           RESET ORDER FLAGS
D707 7B0C 7B0C  DIRO000 XC     #HIGHMEM,#HIGHMEM    CLEAR HIGH MEMBER NAME
D707 7B14 7B14          XC     #DUPLMEM,#DUPLMEM    CLEAR DUPLICATE MEMBER
D203 7B18 834E          MVC    #DUPLMEM+4(4),DIRO0000 INITIAL VALUE FOR ZERO
D707 7D10 7D10          XC     MEMNAME,MEMNAME      CLEAR DUPLICATE MEMBER
9201 7CDE               MVI    STARTTR+2,X'01'      FIRST TTR
D207 7B1C 7D10  DIRO010 MVC    #LASTMEM,MEMNAME     PREVIOUS MEMBER NAME
45E0 7DEC               BAL    R14,READDIR          READ NEXT MEMBER
4110 CCB8               LA     R1,L762$1            MEMBER DUPLICATE MESSA
                        M$MSG (1)                   OUTPUT ONE OF THE MESS
45E0 7DDC       +       BAL    R14,$TMSGRT
4110 C6ED               LA     R1,PDS397A           CORRECT DATA SET?
4520 7E40               BAL    R2,YESNO             PROMPT FOR RESULTS
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
----------------- ISPMODE Session# 1 Log -------- ROW 1,000 TO 1,016 OF 1,016
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=IEFBR14  --------------------------
>----->list iefbr14 dump skiprec(3)
PDS140I DUMP RECORD      4   LENGTH     21   TTR 00120D
  000135 0000  80140480 0200F5F7  F3F4C1E2 F1F0F040  *......5734AS100 *
  000145 0010  06018124 0F                           *..a..*

PDS140I DUMP RECORD      5   LENGTH     20   TTR 00120E
  00014A 0000  80138800 0281336F  0BD9E2C9 F1F2F4F0  *..h..a.?.RSI1240*
  00015A 0010  F0F7F7F5                              *0775*

PDS140I DUMP RECORD      6   LENGTH     20   TTR 00120F
  00015E 0000  0D000000 00040000  06000000 40000008  *............ ...*
  00016E 0010  00020008                              *....*

PDS140I DUMP RECORD      7   LENGTH      8   TTR 001210
  000172 0000  1BFF07FE 00000000                     *........*

PDS142I 7 blocks in this member
****************************** BOTTOM OF DATA ******************************
```

```
 FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
---------------- ISPMODE Session# 1 Log# 1 ---------  ROW 131 TO 156 OF 261
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.VSAM.RV.CLUSTER,VOL=SER=SER007 ----------------------------------
>----->list block
PDS140I BLOCK RECORD        1   LENGTH    29   RBA 00000004
0000001 0000 *0000XX00 SET ZCTVERB  = PDSSX*

PDS140I BLOCK RECORD        2   LENGTH    15   RBA 00000037
0000002 0000 *00000100 PROC 0*

PDS140I BLOCK RECORD        3   LENGTH    31   RBA 00000056
0000003 0000 *00000200  ISPEBEX TBBOT ISPCMDS*

PDS140I BLOCK RECORD        4   LENGTH    57   RBA 00000091
0000005 0000 *00000500 SET ZCTACT   = &STR(SELECT CMD(%PDSSX DUMMYDSN))*

PDS140I BLOCK RECORD        5   LENGTH    49   RBA 00000152
0000006 0000 *00000600  SET ZCTDESC  = 'INVOKE THE PDS PROGRAM'*

PDS140I BLOCK RECORD        6   LENGTH    31   RBA 00000205
0000007 0000 *00000700  ISPEBEX TBADE ISPCMDS*
```

```
 FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------- ISPMODE Session# 1 Log# 1  ------ ROW 675 TO 700 OF 1,075
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007 ----------------------------------
>----->list block skiprec(21)
PDS140I BLOCK RECORD       22   LENGTH    80   RBA 00001484
00000600 000 *00000600A004E   /*   SECONDARY OF THE REQUESTED SIZE FOR DATA S*
00000600 040 *ETS      */                                                    *

PDS140I BLOCK RECORD       23   LENGTH 7,623   RBA 00024576
000007MM 000 *000007MM Format: IDC3351I * VSAM  OPEN | CLOSE | I/O  RETURN   *
000007MM 040 *CODE IS rc                                                     *
000007MM 080 *               RPLFDBWD = nnnnnnnn                             *
000007MM 0C0 *                                                               *
000007MM 100 *                                                               *
000007MM 140 *                                          Description: An e*
000007MM 180 *rror occurred during VSAM processing that resulted in the     *
000007MM 1C0 *                                               return code*
000007MM 200 * identified by 'rc' in the message text. The                  *
000007MM 240 *                                          RPLFDBWD value, if*
000007MM 280 * present, is documented in DFP MACRO IN-                       *
000007MM 2C0 *                                          STRUCTIONS.  The values o*
```

LISTA Function

LA or LISTA command displays your TSO session allocations in an ISPF table. DDNAME is another way to use the LISTA table. The LISTA/DDNAME table is maintained in DDNAME and concatenation order (the SORT command is not supported).

Example

```
LISTA   open  old  mod
```

Syntax

```
LISTA   [ALL            ]
        [OPEN           ]
        [UNOPEN|CLOSED   ]
        [CONCAT|CATENATE ]
        [NONCAT|NONCON   ]
        [DUMMY|NULLFILE  ]
        [               ]
        [TERMINAL        ]
        [JESFILE|SYSOUT  ]
        [TMP|TEMPORY     ]
        [NEW MOD SHR OLD ]
        [TASKLIB        ]
        [SETDEF         ]
        [RESET          ]
```

Aliases

```
LA, LISTA
```

Defaults

ALL

Required

Operands

ALL	Selects all allocations (initial default).
OPEN	Selects open allocations.
UNOPEN	Selects closed allocations.
CLOSED	Selects closed allocations.
CONCAT	Selects allocations that are a part of a concatenation.
CATENATE	Selects allocations that are a part of a concatenation.
NONCAT	Selects allocations that are not concatenated.
NONCON	Selects allocations that are not concatenated.
DUMMY	Selects NULLFILE allocations.
NULLFILE	Selects NULLFILE allocations.
	Selects NULLFILE allocations that are not also TERMINAL.
TERMINAL	Selects TERMINAL allocations.

JESFILE	Selects allocations to JES output files.
SYSOUT	Selects allocations to JES output files.
TMP	Selects temporary allocations.
TEMPORY	Selects temporary allocations.
NEW	Selects NEW allocations.
MOD	Selects MOD allocations.
SHR	Selects SHR allocations.
OLD	Selects OLD allocations.
TASKLIB	Selects TASKLIB allocations (including STEPLIB).
SETDEF	Makes the selected operands the new default for this session instead of ALL.
RESET	Clears the LISTA table before adding any new entries.

Remarks

The LISTA/DDNAME ISPF table displays in response to a LISTA command. When you are in a LISTA/DDNAME display, you can delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, etc. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as either a primary command or as a line command.

LISTA has alternate panels, as do most of the ISPMODE tables in StarTool FDM, containing extra information about the listed data sets. The first LISTA/DDNAME panel shows basic allocation information such as DDNAME, data set TYPE, Open count (under the header O#), VOLUME allocated, MEMBER name, and DSNAME. Command error messages are provided under the DATA/MSG header as follows:

*INVALID	This line command is not supported.
*DD GONE	This file is not allocated.
*OPEN DD	This file is open.
*IN USE	This file is in use.
*NOT CON	This file is not concatenated.
*PERMCON	This file is permanently concatenated.
NO DDN	This file has a blank ddname.

TYPE is the data set DSORG or VIO, CTLG, VVDS, JES, or TERM depending on the type of allocation.

The second LISTA/DDNAME panel is a double line panel that includes some additional information: DISP for data set disposition and STAT for data set status. The STAT field is TMP for temporary, PRM for permanent, or CNV for convertible.

The following primary commands are supported directly for the LISTA/DDNAME function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ALT[ERNAT]	Displays an alternate view of the LISTA/DDNAME table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
BAT[CHJCL]	Builds batch JCL that includes all DD statements in the LISTA table.	
COMPR[ESS]	Global command. Changes to each data set in the table and issues a COMPRESS subcommand.	
	Syntax	`COMPRESS [LIST|NOLIST|SUMMARY]`
DUA[L]	Displays a double line view of the LISTA/DDNAME table.	
DD[NAME]	Rebuilds the LISTA/DDNAME table selecting data sets with a DDNAME mask.	
	Syntax	`DDNAME ddnamemask`
EDITT[BL] (or ET[BL])	Enters an edit session on LISTA/DDNAME table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax	`F anystring [ASIS]`
		`    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
		`    [ANY|DDNAME|DSNAME|VOLUME|MEMBER|ALLOC]`

FI[ND]	Global command. Changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is applied to each data set individually, do not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.	

	Syntax
	`FIND memgroup 'anystring'`
	`    [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`    [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`	

GL[OBAL]	Global command. Changes to each data set in the table and issues a StarTool FDM subcommand.	
	Syntax	`GLOBAL anysubcommand [anyoperands]`
LE[FT]	Rotates through alternate views of the LISTA/DDNAME table. PF keys 10 and 22 are normally set to LEFT.	
LISTA	Rebuilds the LISTA/DDNAME table selecting data sets by attributes.	
	Syntax	`LISTA [attributes]`
MODEL	Global command. Changes to each data set in the table and issues a MODEL command.	
NOR[MAL]	Displays the default view of the LISTA/DDNAME table.	
OUT[PUT]	Outputs the LISTA/DDNAME table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`

REM[OVE]	Trims the LISTA/DDNAME table based on a string match.

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [ANY|DDNAME|DSNAME|VOLUME|MEMBER|ALLOC]`

REPL[ACE]	Global command. Changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is applied to each data set individually, do not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.

	Syntax
	`REPLACE memgroup 'fromstring' 'tostring'`
	`        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`        [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`
	`        [WRITE|NOWRITE]`

RESET	Often used with data set tagging, clears the DATA/MSG field in all table entries.
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.
RI[GHT]	Rotates through alternate views of the LISTA/DDNAME table. PF keys 11 and 23 are normally set to RIGHT.
SEEK	Global command. Changes to each data set in the table and looks for a member.
	Syntax	`SEEK member`
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.
X	Clears the LISTA/DDNAME table relative to the cursor position.
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the LISTA/DDNAME table; this is equivalent to X ALL

The following line commands are supported in the LISTA/DDNAME function:

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
6	Changes to the data set and issues a MEMLIST all.
A	Changes to the data set and executes an alternate subcommand.
B	Changes to the data set and issues a MEMLIST all.
C	Changes to the data set.
COPY	Changes to the data set and invokes the COPY subcommand.
DCAT	Deconcatenates non-permanent data sets in a concatenated group.
DUP	Changes to the data set and invokes the DUP subcommand.

E	Changes to the data set and issues a MEMLIST all.
FREE	Frees a DDNAME or all data sets in a concatenated group.
GO	Changes to the data set using GO processing (a number is optional).
IDC	Performs an IDCAMS LISTC and directs output to the log.
K	Kills and clears all following line commands.
LC	Adds the data set name to the current LISTC/LISTF table.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST all.
MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
S	Selects line command (normally CHANGE, it is set by SETSEL).
SEEK	Changes to the data set and checks for a member.
SETA	Saves a StarTool FDM subcommand for repeated use by the ALT line command.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
U	Changes to the data set and displays USAGE.
UT	Selects the extended user line command panel.
W	Adds the data set name (and any member name) to the WORKPAD.
WHO	Checks for users of this data set with the WHOHAS subcommand.
X	Drops the table line.
XCOP	Changes to the data set and invokes the XCOPY interface.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTA/DDNAME function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR- t="Block line command,==">`
AA	A command. Changes and executes an alternate subcommand for each line.
BB	B command. Changes and issues a MEMLIST all for each line.
CC	C command. Changes data set for each line.
EE	E command. Changes and issues a MEMLIST all for each line.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	M command. Changes and issues a MEMLIST all for each line.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
SSEE	SEEK command. Changes to each data set and checks for a member.

UU	U command. Changes and displays USAGE for each line.
XX	X command. Drops the range of table lines.
ZZ	Z command. Changes and issues a COMPRESS for each line.

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS FEATURE
---------------------- List Allocations -----------------    ROW 1 TO 7 OF 39
COMMAND ===>                                              SCROLL ==> CSR
- DSN=SER07.LIB.CLISTV,VOL=SER=SER006  MEM=/AXI  -----------------------------
CMD  DDNAME    DATA/MSG I/O's O# VOLUME  MEMBER    ------ DATA SET NAME --------
     STEPLIB          34 1 SER007           SER07.LINK.LOAD
       #    2           0 1 SER007           SER09.SDS.LOAD
       #    3           0 1 SER007           SER.COMPAREX.LINKLIB
     ISPPLIB         195 1 SER006           SER07.LIB.PDSE
       #    2           2 1 SER007           SER.COMPAREX.PANELS
       #    3           0 1 TSG303           SYSI.IBMPDF.SYS350.PREPLIB
       #    4          27 1 TSG302           SYSI.IBMPDF.SYS350.ISRENU
       #    5           0 1 TSG309           SYSI.IBMPDF.SYS350.ISPENU
       #    6           0 1 TSG301           SYSP.CUST.ISPPLIB
     ISPMLIB           4 1 SER006           SER07.LIB.PDSE
o      #    2           0 1 SER007           SER.COMPAREX.MSGS
       #    3           3 1 TSG302           SYSI.IBMPDF.SYS350.ISRENU
       #    4          99 1 TSG309           SYSI.IBMPDF.SYS350.ISPENU
       #    5           0 1 TSG301           SYSP.STD.ISPMLIB
     ISPSLIB           0 1 TSG301           SYSP.STD.ISPSLIB
       #    2           0 1 TSG302           SYSI.IBMPDF.SYS350.ISRENU
     ISPTLIB           1 1 TSG301           SYSP.STD.ISPTLIB
```

```
--------------------- StarTool o LISTA Line Command ------------------------
OPTION  ===>

Choose one of the following line commands:
                                                      More:     +
    +    - provide an extension panel for command entry.
    =    - repeat the previous line command.
    6    - change to the data set and MEMLIST all.
    a    - change to the data set and execute an alternate subcommand.
    b    - change to the data set and MEMLIST all.
    c    - change to 'SER.COMPAREX.MSGS'.
    dcat - deconcatenate non-permanent members in a concatenated group.
    e    - change to the data set and MEMLIST all.
    free - free a ddname or all data sets in a concatenated group.
    go   - change to the data set using GO processing (a number is optional).
    idc  - perform an IDCAMS LISTC with output to log.
    k    - kill and clear all following line commands.
    log  - format and copy line to log.
    m    - change to the data set and MEMLIST all.
    mod  - change to the data set and MODEL.
    s    - SELECT line command (normally CHANGE, it is set by SETSEL).
    seek - change to the data set and check for a member.
    u    - change to the data set and display USAGE.
    ut   - use the extended user line command panel.
```

LISTC Function

Use the LC or LISTC command to display data sets from a catalog source in an ISPF table. The table displaying LISTC data sets is combined with LISTF data sets.

LISTC/LISTF tables can be saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE, and ERASE commands. To automatically save these tables at program end or when a new LISTC/LISTF table is retrieved, check the setting of "Automatic save" in SETALL for LISTC/LISTF.

Saved LISTC/LISTF tables are given members names of the form PDSLF0na if the name is one or two numeric digits; otherwise, a name of the form ##name is used where name is a 1- to 6-character alphanumeric table identification name.

The FIND and REPLACE global subcommands bypass non-partitioned data sets. If you want to search or update non-partitioned data sets, use GLOBAL followed by FIND or REPLACE and any operands.

In the LISTC prompt panel, enter the LEVEL/DSN field as a partial data set name. The first level can contain an ***** after one or more characters and any additional levels can contain an ***** before or after any characters in that level. For example: SYS*.NU*.*.*XY

Due to a restriction in the SUPERLOCATE service, generic searches for GDG data sets must contain an * in the first position of the GDG data set node. For example, if you search for NODE1.GDG.G00*, no GDG entries will be found. However, you can find these entries by changing the search to NODE1.GDG.*G00.

You can bypass the LISTC or LISTF prompting panels by using the MASK command. See "MASK Function".

Example

```
LISTC
```

Syntax

```
LISTC [name] [RESET] [PROMPT]
```

Aliases

```
LC, LISTC
```

Defaults

none – add to the LISTC/LISTF table

Required

Operands

name	1- to 6-alphanumeric characters, the LISTC/LISTF table name to retrieve.
RESET	Clears the LISTC/LISTF table before adding any new entries.
PROMPT	Prompts for LISTC operands even if a LISTC/LISTF table exists.

Remarks

The LISTC/LISTF ISPF table displays in response to a LISTC command. When you are in a LISTC display, you can delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, etc. For assistance with the different options available, use the HELP command, CUA functions, or you enter an O command as either a primary command or as a line command.

There are ten defined formats for LISTC/LISTF. Select which panels you want to use with the SETPANEL command as shown below. You can switch between these panel formats with RIGHT and LEFT commands. Following this panel is a sample of each of the ten defined panel formats. StarTool FDM displays DSORG (under header DO) as DA for direct, PE for partitioned extended, PO for partitioned, PS for sequential, and VS for VSAM.

Each of the panels contain a column of data under the C (for Catalog) header which is Y if the data set is cataloged, N if the data set is not cataloged, and - if the data set catalog status was not checked.

The column of data under the V (for Volume) header provides feedback as to whether or not the data set is present on the volume. Y and N are short for YES or NO, M is short for Managed (meaning that it is a SMS data set), and - means that the status of the data set was not checked.

The following primary commands are supported directly for the LISTC function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ALT[ERNAT]	Displays an alternate view of the LISTC/LISTF table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
COMPR[ESS]	Global command. Changes to each data set in the table and issues a COMPRESS subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.	
	Syntax	`COMPRESS [LIST|NOLIST|SUMMARY]`
DUA[L]	Displays a triple line view of the LISTC/LISTF table.	
EDITT[BL] (or ET[BL])	Enters an edit session on LISTC/LISTF table data.	
ER[ASE]	Deletes the LISTC/LISTF table in memory and on disk.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax
	``
	F anystring [ASIS]
	   [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]
	   [ANY|DSNAME|VOLUME|DO|RECFM|CDATE| RDATE]
	``

FI[ND]	Global command. Changes to each table data set and issues a FIND subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.

	Syntax
	``
	FIND memgroup 'anystring'
	     [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]
	     [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]
	``

GL[OBAL]	Global command. Changes to each data set in the table and issues a StarTool FDM subcommand.

	Syntax	```GLOBAL anysubcommand [anyoperands]```
ID	Changes the table ID name.	
	Syntax	```ID name```
	(where *name* is a 1- to 6-character alphanumeric name).	
LE[FT]	Rotates through alternate views of the LISTC/LISTF table. PF keys 10 and 22 are normally set to LEFT.	
LISTC	Searches for more data sets from a catalog source (after prompting).	
	Syntax	```LISTC [name] [RESET]```
	(where name is 1- to 6-alphanumeric characters)	
LISTF	Searches for more data sets from a volume source (after prompting).	
	Syntax	```LISTF [name] [RESET]```
	(where name is 1- to 6-alphanumeric characters)	
LO[AD]	Loads data set names from operating system control blocks.	
	Syntax	```LOAD APFLST|LNKLST|LPALST [RESET]```
	LOAD LNKLST now supports both static and dynamic linklists.	
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	

	Syntax	```
		LOCATE data
		          pdsname|pvolume|cdate|rdate|trksize|
		          trkfree|extents|blksize|dsorg|alloc
		``

MER[GE]	Merges data from a saved LISTC/LISTF table into the current table.	
	Syntax	```MERGE name [GROUP|SAMPLE] [RESET]```
	(where name is 1- to 6-alphanumeric characters)	
MODEL	Global command. Changes to each data set in the table and issues a MODEL command.	
NOR[MAL]	Displays the default view of the LISTC/LISTF table.	
NOS[AVE]	Specifies that the current LISTC/LISTF table should not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF.	

O[PTIONS]	Provides primary command selection for the LISTC/LISTF function and operand syntax assistance.	
OUT[PUT]	Outputs the LISTC/LISTF table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REF[RESH]	Global command. Updates information and status for all lines in the LISTC/LISTF table. Also, REFRESH corrects volume name entries for cataloged data sets.	
REM[OVE]	Trims the LISTC/LISTF table based on a string match.	

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [ANY|DSNAME|VOLUME|DO|RECFM|CDATE|RDATE]`	

REPL[ACE]	Global command. Changes to each data set in the table and issues a REPLACE subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.	

	Syntax
	`REPLACE memgroup 'fromstring' 'tostring'`
	`        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`        [CAPS|ASIS|IGNORE]   [WORD|PREFIX|SUFFIX]`
	`        [WRITE|NOWRITE]`	

RESET	Often used with data set tagging, clears the DATA/MSG field in all table entries.	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the LISTC/LISTF table. PF keys 11 and 23 are normally set to RIGHT.	
SAVE	Creates a permanent table for use in a different StarTool FDM session.	
	Syntax	`SAVE [name] [REPLACE|NOREPL]`
	(where name is 1 to 6 alphanumeric characters)	
SEEK	Global command. Changes to each data set in the table and looks for a member. In LISTC/LISTF tables, non-partitioned data sets are skipped.	
	Syntax	`SEEK member`
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	

	Syntax	`SORT [DSNAME|VOLUME|CDATE|RDATE|SIZE|FREE`
		`        |EXTENTS|BLKSIZE|TYPE|ALLOC]`
		`     [ASCEND|DESCEND]`

TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.	
WHO[HAS]	Global command. Issues a WHOHAS subcommand for each data set in the table.	

X	Clears the LISTC/LISTF table relative to the cursor position.
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the LISTC/LISTF table; this is equivalent to X ALL

The following line commands are supported in the LISTC function – note that the DFHSM commands are supported from the UT panel.

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
6	Changes to the data set and issues a MEMLIST all.
A	Changes to the data set and executes an alternate subcommand.
ACAT	Catalogs a non-VSAM data set using an alternate user catalog from the UT panel.
AUNC	Uncatalogs a non-VSAM data set using an alternate user catalog from the UT panel.
B	Changes to the data set and issues a MEMLIST all.
C	Changes to the data set.
CAT	Catalogs this data set.
COPY	Changes to the data set and invokes the COPY subcommand.
DEL	Deletes this data set (normally after a confirmation prompt).
DUP	Changes to the data set and invokes the DUP subcommand.
E	Changes to the data set and issues a MEMLIST all.
GO	Changes to the data set using GO processing (a number is optional).
HBAC	HBBACK DFHSM command to back up the data set.
HBDE	HBDEL DFHSM command for the data set.
HDEL	HDEL DFHSM command for the data set.
HMIG	HMIG DFHSM command for the data set.
HML2	HMIG DFHSM command for the data set to LEVEL 2.
HREC	HRECALL DFHSM command for the data set.
HSM	Prompts for various HSM actions on a data set.
IDC	Performs an IDCAMS LISTC and directs output to the log.
INFO	Displays information on the data set in panel format; also, updates data set information.
K	Kills and clears all following line commands.
LC	Checks the catalog entry for this data set and merges that entry into the LISTC/LISTF table (if different); also, updates data set information.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST all.

MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
PB	Changes to the data set and PBROWSEs this data set.
PE	Changes to the data set and PEDITs this data set.
RACF	Provides entry assist panels for RACF commands LISTDSD, ADDSD, PERMIT, ALTDSD, DELDSD, and SEARCH.
REFR	Refreshes and updates information and status for this data set.
REN	Renames this data set (after prompting).
S	Selects line command (normally CHANGE, it is set by SETSEL).
SEEK	Changes to the data set and checks for a member.
SETA	Saves a StarTool FDM subcommand for repeated use by the ALT line command.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
U	Changes to the data set and displays USAGE.
UNC	Uncatalogs this data set (not allowed for SMS-managed data sets).
UT	Selects the extended user line command panel.
VSAM	Invokes VSAM manipulation and information services.
W	Adds the data set name to the WORKPAD.
WHO	Checks for users of this data set with the WHOHAS subcommand.
X	Drops the table line.
XCOP	Changes to the data set and interfaces with extended copy.
XMIT	Transmits the data set to another user.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTC function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR- t="Block line command,==">`
AA	A command. Changes and executes an alternate subcommand for each line.
BB	B command. Changes and issues a MEMLIST all for each line.
CC	C command. Changes data set for each line.
DDEL	DEL command. Deletes each data set (normally after a confirmation prompt on each one).
EE	E command. Changes and issues a MEMLIST all for each line.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	ML command. Changes and issues a MEMLIST all for each line.
OO	O command. Provides line command assistance for each line.

SS	S command. Selects each line in the range of table lines.
SSEE	SEEK command. Changes to each data set and checks for a member.
UU	U command. Changes and displays USAGE for each line.
XX	X command. Drops the range of table lines.
ZZ	Z command. Changes and issues a COMPRESS for each line.

```
--------------------------- Set PANEL Defaults ----------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
                                                        More:     +
LISTC/LISTF Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES  (Yes/No) -- VOLUME DO RECFM LRECL BLKSI
 Show Dsname view      ===> NO   (Yes/No) -- VOLUME DO RECFM (full dsname)
 Show Custom view      ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Size view        ===> YES  (Yes/No) -- SIZE FREE USED DEVICE
 Show Extent view      ===> YES  (Yes/No) -- EXT SEC ALLOC RND KEY RKP
 Show User view        ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Double line view ===> YES  (Yes/No) -- VOLUME DEVICE DO .. (for 2 lines
 Show Created view     ===> NO   (Yes/No) -- CREATED REFERENCED UPD
 Show Expiration view  ===> NO   (Yes/No) -- EXPIRATION DIR USED RACF
 Show Total view       ===> NO   (Yes/No) -- VOLUME DEVICE DO .. (for 3 lines)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment      ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment      ===> 50       volumes processed between messages
 VMAP increment         ===> 500      data sets processed between messages
```

```
-------------------- List files TESTXX - (Attributes) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB      80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB     255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
---------------------- List files TESTXX - (Dsname) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DO RECFM
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
--------------------- List files TESTXX - (Custom) --------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB      80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB     255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
---------------------- List files TESTXX - (Size) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ----------   SIZE    FREE USED DEVICE
    Y Y *CHANGE  WSER07.LIB.CLIST_____   100T     4T  96% 3380K
    Y Y *REFRESH WSER07.LIB.CLISTV_____     2T     0T 100% 3390M3
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
---------------------- List files TESTXX - (Extent) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXT   SEC ALLOC RND KEY RKP
    Y Y *CHANGE  WSER07.LIB.CLIST_____   3    33 TRK   NO   0   0
    Y Y *REFRESH WSER07.LIB.CLISTV_____   1    10 TRK   NO   0   0
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
--------------------- List files TESTXX - (User) ----------  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
-------------------- List files TESTXX - (Double line) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DEVICE DO
      RECFM LRECL BLKSI    SIZE    FREE USED CREATED    EXPIRATION REFERENCED
    Y Y *CHANGE* WSER07.LIB.CLIST_____ STR911 3380K  PO
      FB       80  9040    100T     4T  96% 1989/10/04 ** NONE ** 1995/06/28
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P 3390M3 PO
      VB      255  9040      2T     0T 100% 1995/03/16 1995/12/31 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
---------------------- List files TESTXX - (Created) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- CREATED    REFERENCED UPD
    Y Y *CHANGE  WSER07.LIB.CLIST_____ 1989/10/04 1995/06/28 U
    Y Y *REFRESH WSER07.LIB.CLISTV_____ 1994/03/16 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
--------------------- List files TESTXX - (Expiration) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXPIRATION   DIR  USED RACF
    Y Y *CHANGE* WSER07.LIB.CLIST_____ ** NONE **    73    64
    Y Y *REFRESH WSER07.LIB.CLISTV_____ 1995/12/31
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
----------------------- List files TESTXX - (Total) ------- ROW 1 TO 3 OF 3
COMMAND ===>                                              SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75 -------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DEVICE
          DO    LRECL KEY CREATED  EXPIRATION   --SIZE- -DIR- ALLOC USED
               RECFM BLKSI  RKP  REFERENCED UPD EXT   --FREE- -USED -SEC-  RND
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 3380K
          PO     80   0  1989/10/04 ** NONE **    100T        TRK   96%
          FB    9040   0  1995/06/28 U    3          4T              33  NO
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P 3390M3
          PO    255   0  1994/03/16 1994/12/31     2T         TRK  100%
          FB    9040   0  1995/06/20       1        0T              10  NO
    Y - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
--------------------------- ut for LC   user line commands ------------------
OPTION  ===>


Choose one of the following:
    ALC  - Allocate data set prompt
    HDEL - HDEL command for data set 'SER07.SMSTEST.DATA9040'
    HMIG - HMIG command for data set 'SER07.SMSTEST.DATA9040'
    HML2 - HMIG to LEV2 for data set 'SER07.SMSTEST.DATA9040'
    HBDE - HBDEL command for data set 'SER07.SMSTEST.DATA9040'
    HREC - HRECALL command for data set 'SER07.SMSTEST.DATA9040'

Dynamic Commands--note:  < = MVSD4B,
                         / = 'SER07.SMSTEST.DATA9040'
DSN3 - CMD(DSAT / VOL(<))    /* THIS IS A COMMENT
LCLC - TRP(LISTC ENT(/) ALL) /* THIS IS A COMMENT
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
____ -
```

```
------------------------ Dataset Information Display ----------------------
OPTION ===>


 Dataset name:  SER07.SMSTEST.DATA08
    on Volume:  MVSD4B                          SMS classes
 DSORG:  PE       CREATION DATE:   1992/07/31 STORAGE CLASS:    TESTSC
 RECFM:  FB       REFERENCE DATE:  1993/04/14 MANAGEMENT CLASS: (NULL)
 LRECL:  80       EXPIRATION DATE: 1992/04/09 DATA CLASS:       DATAF
 BLKSIZE: 32720   UPDATE FLAG:     U          Last backup:      00.000
                                              DATASET TYPE:     LIBRARY
 ALLOCATED SPACE: 132T
 UNUSED SPACE:   0T              KEY LENGTH:   0
 ALLOCATION TYPE: TRK           KEY POSITION: 0
 SECONDARY SIZE:  1
 EXTENT COUNT:    123
```

```
-------------------------- VSAM KSDS information --------------------------
OPTION  ===>
                                                              More:    +
 Cluster name ===> SER07.VSAM.IS.CLUSTER
 Data    name ===> SER07.VSAM.IS.CLUSTER.DATA
 Index   name ===> SER07.VSAM.IS.CLUSTER.INDEX
 Catalog name ===> ICFUCAT.VTSG312
 Key length   ===>        8         Owner         ===> (NULL)
 Key offset   ===>        0         Creation Date ===> 1993.006
 Average LRECL===>      200         Expiration    ===> 1994.164  14.24.07
 Maximum LRECL===>     9000
     SPND      NERAS    NRUS        Storage class ===>
     UNORD              SHR(1,3)    Management    ===>
                                    Data class    ===>
 Data space usage  -TRACKS-   KILOBYTES    - CA's -    - CI's -    percent
 Allocated space         1         40           1          10
 High used space         1         40           1          10      100.0
 Real used space         1         28           1           7       70.0
 Index space usage -TRACKS-   KILOBYTES    - CA's -    - CI's -    percent
 Allocated space         1         40           1          10
 High used space         1          4           1           1       10.0


 Usage indicators for space and performance


     NIMBD     NREPL    SPEED
 Space indicators  -- CI -- -- CA --        Usage indicators
 Percent free ===>      10       10    Total records ===>      80
 Splits       ===>       2        0    Deleted       ===>       0
 Percent Splits >     20.0      0.0    Updated       ===>       8
 attributes       - Data - - Index-    Inserted      ===>       8
 CI Size      ===>     4096     4096    Retrieved     ===>    4549
 CIs per CA   ===>       10       10
 Block size   ===>     4096     4096    Buffer space  ===>   12288
 Block/Track  ===>       10       10    Index levels  ===>       1
 Tracks/CA    ===>        1        1    Index records ===>       1
```

LISTF Function

Use the LF or LISTF command to display data sets from a volume or VTOC source in an ISPF table. The table displaying LISTF data sets is combined with LISTC data sets.

LISTC/LISTF tables can be saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE, and ERASE commands. To automatically save these tables at program end or when a new LISTC/LISTF table is retrieved, check the setting of "Automatic save" in SETALL for LISTC/LISTF.

Saved LISTC/LISTF tables are given member names of the form PDSLF0na if the name is one or two numeric digits; otherwise, use a name of the form ##name where *name* is a 1- to 6-character alphanumeric table identification name.

In the LISTF prompt panel, enter a volume mask in the search field and the LEVEL/DSN field as a partial data set name. For the LISTF volume mask, use a **/** may for a pattern specification. Use an ***** for a combination name, and use a **:** for a range. For example: TSO**/**03, TSO*, and TSO:TSO80. For the data set name, each level can contain an ***** before or after any characters in that level. For example: SYS*.NU*.*.*XY.

You can bypass the LISTC or LISTF prompting panels by using the MASK command. See "MASK Function".

Example

```
LISTF
```

Syntax

```
LISTF [name] [RESET] [PROMPT]
```

Aliases

```
LF, LISTF
```

Defaults

none—add to the LISTC/LISTF table

Required

Operands

name	1- to 6-alphanumeric characters, the LISTC/LISTF table name to retrieve.
RESET	Clears the LISTC/LISTF table before adding any new entries.
PROMPT	Prompts for LISTF operands even if a LISTC/LISTF table exists.

Remarks

The LISTF/LISTC ISPF table displays in response to a LISTF command. When you are in a LISTF display, you can delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, etc. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as either a primary command or as a line command.

There are ten defined formats for LISTC/LISTF. Select which panels you want to use with the SETPANEL command, as shown below. You can switch between these panel formats with RIGHT and LEFT commands. Following this panel is a sample of each of the ten defined panel formats. StarTool FDM displays DSORG (under header DO) as DA for direct, PE for partitioned extended, PO for partitioned, PS for sequential, and VS for VSAM.

Each of the panels contain a column of data under the C (for Catalog) header which is Y if the data set is cataloged, N if the data set is not cataloged, and - if catalog status was not checked.

The column of data under the V (for Volume) header provides feedback as to whether or not the data set is present on the volume. Y and N are short for YES or NO, M is short for

Managed (meaning that it is a SMS data set), and - means that the status of the data set was not checked.

The following primary commands are supported directly for the LISTF function; for information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ALT[ERNAT]	Displays an alternate view of the LISTC/LISTF table.
APP[LY]	Applies the specified line command to all table entries and executes each entry.
	Syntax `APPLY linecmd`
COMPR[ESS]	Global command. Changes to each data set in the table and issues a COMPRESS subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.
	Syntax `COMPRESS [LIST|NOLIST|SUMMARY]`
DUA[L]	Displays a triple line view of the LISTC/LISTF table.
EDITT[BL] (ET[BL])	Enters an edit session on LISTC/LISTF table data.
ER[ASE]	Deletes the LISTC/LISTF table in memory and on disk.
EXPR[ESS]	Executes all entered line commands without pauses between commands.
F	Finds a string and positions the display start location.

	Syntax `F anystring [ASIS]`
	`    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	`    [ANY|DSNAME|VOLUME|DO|RECFM|CDATE|RDATE]`

FI[ND]	Global command. Changes to each table data set and issues a FIND subcommand. In LISTC/LISTF tables, non-partitioned data sets are skipped.

	Syntax `FIND memgroup 'anystring'`
	`    [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`    [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`

GL[OBAL]	Global command. Changes to each data set in the table and issues a StarTool FDM subcommand.
	Syntax `GLOBAL anysubcommand [anyoperands]`
ID	Changes the table ID name.
	Syntax `ID name`
	(where *name* is a 1- to 6-character alphanumeric name)
LE[FT]	Rotates through alternate views of the LISTC/LISTF table. PF keys 10 and 22 are normally set to LEFT.
LISTC	Searches for more data sets from a catalog source (after prompting).
	Syntax `LISTC [name] [RESET]`
	(where *name* is 1- to 6-alphanumeric characters)
LISTF	Searches for more data sets from a volume source (after prompting).
	Syntax `LISTF [name] [RESET]`

	(where *name* is 1- to 6-alphanumeric characters)	
LO[AD]	Loads data set names from operating system control blocks.	
	Syntax	LOAD APFLST|LNKLST|LPALST [RESET]
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	

	Syntax	`LOCATE name`
		`        pdsname|pvolume|cdate|rdate|trksize|`
		trkfree|extents|blksize|dsorg|alloc

MER[GE]	Merges data from a saved LISTC/LISTF table into the current table.	
	Syntax	MERGE name [GROUP|SAMPLE] [RESET]
	(where *name* is 1- to 6-alphanumeric characters)	
MODEL	Global command. Changes to each data set in the table and issues a MODEL command.	
NOR[MAL]	Displays the default view of the LISTC/LISTF table.	
NOS[AVE]	Specifies that the current LISTC/LISTF table not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF.	
O[PTIONS]	Provides primary command selection for the LISTC/LISTF function and operand syntax assistance.	
OUT[PUT]	Outputs the LISTC/LISTF table to print or a data set.	
	Syntax	OUTPUT [=c | F(ddname)]
REF[RESH]	Global command. Updates information and status for all lines in the LISTC/LISTF table. Also, REFRESH corrects volume name entries for cataloged data sets.	
REM[OVE]	Trims the LISTC/LISTF table based on a string match.	

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [ANY|DSNAME|VOLUME|DO|RECFM|CDATE|RDATE]`	

REPL[ACE]	Global command. Changes to each data set in the table and issues a REPLACE command. In LISTC/LISTF tables, non-partitioned data sets are skipped.	

	Syntax
	`REPLACE memgroup 'fromstring' 'tostring'`
	`        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`        [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`
	`        [WRITE|NOWRITE]`	

RESET	Often used with data set tagging, clears the DATA/MSG field in all table entries.	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the LISTC/LISTF table. PF keys 11 and 23 are normally set to RIGHT.	
SAVE	Creates a permanent table for use in a different StarTool FDM session.	
	Syntax	SAVE [name] [REPLACE|NOREPL]

	(where *name* is 1- to 6-alphanumeric characters)	
SEEK	Global command. Changes to each data set in the table and looks for a member. In LISTC/LISTF tables, non-partitioned data sets are skipped.	
	Syntax	`SEEK member`
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [DSNAME|VOLUME|CDATE|RDATE|SIZE|FREE` `|EXTENTS|BLKSIZE|TYPE|ALLOC]` `[ASCEND|DESCEND]`
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.	
WHO[HAS]	Global command. Issues a WHOHAS subcommand for each data set in the table.	
X	Clears the LISTC/LISTF table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the LISTC/LISTF table; this is equivalent to X ALL	

The following line commands are supported in the LISTF function – note that the DFHSM commands are supported from the UT panel.

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
6	Changes to the data set and issues a MEMLIST all.
A	Changes to the data set and executes an alternate subcommand.
ACAT	VSAM data set using an alternate user catalog from the UT panel.
AUNC	Uncatalogs a non-VSAM data set using an alternate user catalog from the UT panel.
B	Changes to the data set and issues a MEMLIST all.
C	Changes to the data set.
CAT	Catalogs this data set.
COPY	Changes to the data set and invokes the COPY subcommand.
DEL	Deletes this data set (normally after a confirmation prompt).
DUP	Changes to the data set and invokes the DUP subcommand.
E	Changes to the data set and issues a MEMLIST all.
GO	Changes to the data set using GO processing (a number is optional).
HBAC	HBBACK DFHSM command to back up the data set.

HBDE	HBDEL DFHSM command for the data set.
HDEL	HDEL DFHSM command for the data set.
HMIG	HMIG DFHSM command for the data set.
HML2	HMIG DFHSM command for the data set to LEVEL 2.
HREC	HRECALL DFHSM command for the data set.
HSM	Prompts for various HSM actions on a data set.
IDC	Performs an IDCAMS LISTC and directs output to the log.
INFO	Displays information on the data set in panel format; also, updates data set information.
K	Kills and clears all following line commands.
LC	Checks the catalog entry for this data set and merges that entry into the LISTC/LISTF table (if different); also, updates data set information.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST all.
MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
PB	Changes to the data set and PBROWSEs this data set.
PE	Changes to the data set and PEDITs this data set.
RACF	Provides entry assist panels for RACF commands: LISTDSD, ADDSD, PERMIT, ALTDSD, DELDSD, and SEARCH.
REFR	Refreshes and updates information and status for this data set.
REN	Renames this data set (after prompting).
S	Selects line command (normally CHANGE, it is set by SETSEL).
SEEK	Changes to the data set and checks for a member.
SETA	Saves a StarTool FDM subcommand for repeated use by the ALT line command.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
U	Changes to the data set and displays USAGE.
UNC	Uncatalogs this data set (not allowed for SMS-managed data sets).
UT	Selects the extended user line command panel.
VSAM	Invokes VSAM manipulation and information services.
W	Adds the data set name to the WORKPAD.
WHO	Checks for users of this data set with the WHOHAS subcommand.
X	Drops the table line.
XCOP	Changes to the data set and interfaces with extended copy.
XMIT	Transmits the data set to another user.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTF function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR-t="Block line command,==">`
AA	A command. Changes and executes an alternate subcommand for each line.
BB	B command. Changes and issues a MEMLIST all for each line.
CC	C command. Changes data set for each line.
DDEL	DEL command. Deletes each data set (normally after a confirmation prompt on each one).
EE	E command. Changes and issues a MEMLIST all for each line.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	ML command. Changes and issues a MEMLIST all for each line.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
SSEE	SEEK command. Changes to each data set and checks for a member.
UU	U command. Changes and displays USAGE for each line.
XX	X command. Drops the range of table lines.
ZZ	Z command. Changes and issues a COMPRESS for each line.

```
--------------------------- Set PANEL Defaults ----------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
                                                          More:     +
LISTC/LISTF Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES (Yes/No) -- VOLUME DO RECFM LRECL BLKSI
 Show Dsname view      ===> NO  (Yes/No) -- VOLUME DO RECFM (full dsname)
 Show Custom view      ===> NO  (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Size view        ===> YES (Yes/No) -- SIZE FREE USED DEVICE
 Show Extent view      ===> YES (Yes/No) -- EXT SEC ALLOC RND KEY RKP
 Show User view        ===> NO  (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Double line view ===> YES (Yes/No) -- VOLUME DEVICE DO .. (for 2 lines
 Show Created view     ===> NO  (Yes/No) -- CREATED REFERENCED UPD
 Show Expiration view  ===> NO  (Yes/No) -- EXPIRATION DIR USED RACF
 Show Total view       ===> NO  (Yes/No) -- VOLUME DEVICE DO .. (for 3 lines)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500     members processed between messages
 LISTC/LISTF increment ===> 250     data sets processed between messages
 LISTVOL increment     ===> 50      volumes processed between messages
 VMAP increment        ===> 500     data sets processed between messages
```

```
--------------------- List files TESTXX - (Attributes) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB      80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB     255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
---------------------- List files TESTXX - (Dsname) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DO RECFM
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
--------------------- List files TESTXX - (Custom) --------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB      80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB     255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
----------------------- List files TESTXX - (Size) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME ----------   SIZE    FREE USED DEVICE
    Y Y *CHANGE  WSER07.LIB.CLIST_____   100T     4T  96% 3380K
    Y Y *REFRESH WSER07.LIB.CLISTV_____    2T      0T 100% 3390M3
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
---------------------- List files TESTXX - (Extent) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXT   SEC ALLOC RND KEY RKP
    Y Y *CHANGE  WSER07.LIB.CLIST_____  3    33 TRK   NO    0   0
    Y Y *REFRESH WSER07.LIB.CLISTV_____  1    10 TRK   NO    0   0
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
--------------------- List files TESTXX - (User) ----------  ROW 1 TO 3 OF 3
COMMAND ===>                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75 ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 PO FB      80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P PO VB     255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
--------------------- List files TESTXX - (Double line) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                           SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  ------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DEVICE DO
      RECFM LRECL BLKSI   SIZE    FREE USED CREATED    EXPIRATION REFERENCED
    Y Y *CHANGE* WSER07.LIB.CLIST_____ STR911 3380K  PO
       FB     80  9040    100T    4T  96% 1989/10/04 ** NONE ** 1995/06/28
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P 3390M3 PO
       VB    255  9040      2T    0T 100% 1995/03/16 1995/12/31 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
--------------------- List files TESTXX - (Created) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                           SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- CREATED    REFERENCED UPD
    Y Y *CHANGE  WSER07.LIB.CLIST_____ 1989/10/04 1995/06/28 U
    Y Y *REFRESH WSER07.LIB.CLISTV_____ 1994/03/16 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
--------------------- List files TESTXX - (Expiration) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                           SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXPIRATION   DIR   USED RACF
    Y Y *CHANGE* WSER07.LIB.CLIST_____ ** NONE **    73    64
    Y Y *REFRESH WSER07.LIB.CLISTV_____ 1995/12/31
    - - *REFRESH WSER07.LIB.CLISTVV_____
```

```
----------------------- List files TESTXX - (Total) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                           SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  ------------------
CMD C V DATA/MSG ---------DATA SET NAME -------------------- VOLUME DEVICE
         DO    LRECL  KEY  CREATED  EXPIRATION   --SIZE-  -DIR-  ALLOC USED
         RECFM BLKSI  RKP  REFERENCED UPD EXT    --FREE-  -USED  -SEC-  RND
    Y Y *CHANGE  WSER07.LIB.CLIST_____ STR911 3380K
         PO      80    0  1989/10/04 ** NONE **    100T          TRK   96%
         FB    9040    0  1995/06/28 U    3          4T           33   NO
    Y Y *REFRESH WSER07.LIB.CLISTV_____ STR92P 3390M3
         PO     255    0  1994/03/16 1994/12/31      2T          TRK  100%
         FB    9040    0  1995/06/20      1          0T           10   NO
    Y - *REFRESH WSER07.LIB.CLISTVV_____ MIGRAT
```

```
------------------- StarTool o LISTC/LISTF Line Command --------------------
OPTION  ===>

Choose one of the following line commands:
                                                          More:     +
   a    - change to the data set and execute an alternate saved command.
   b    - change to the data set and MEMLIST all.
   c    - change to 'SER07.SMSTEST.DATA9040'
   cat  - catalog the data set.
   del  - delete the data set (after prompting).
   e    - change to the data set and MEMLIST all.
   go   - change to the data set using GO processing (a number is optional).
   idc  - perform an IDCAMS LISTC with output to log.
   info - display data set information in panel format and update statistics.
   lc   - check the catalog entry and merge that information into the table.
   log  - format and copy line to log.
   m    - change to the data set and MEMLIST all.
   ren  - rename the data set (after prompting).
```

LISTGRP Subcommand

The LISTGRP subcommand lists the member group specifications for the current member group.

LISTGRP lists only the names of the member groups; it does not list member names.

Example

LISTGRP

Syntax

LISTGRP

Aliases

LISTG, LISTGR, LISTGRP

Defaults

Required

Operands

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------ ISPMODE Session# 1 Log# 1 ------------  ROW 1 OF 15
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.PDSE112.CLIST,VOL=SER=SER002  MEM=(ADD* ------------------------
>----->AT  (add*,cut/,jcl/)
PDS230I ISPF Stats: VER.MOD  CREATED    LAST MODIFIED  SIZE  INIT   MOD   ID
PDS230I  ADDCYL      01.00  93/05/07  93/05/07  7:55    42    42     0  SER07
PDS230I  ADDDIR      01.02  93/05/07  93/05/07  8:45    48    48     3  SER07
PDS230I  ADDTRK      01.03  93/05/07  93/05/07  8:27    42    42     4  SER07
PDS230I  CUT         01.00  93/06/07  93/06/07  9:29   160   160     0  SER07
PDS230I  CUTJCL      01.22  93/06/18  93/06/20  6:53   179   168     0  SER09
PDS230I  $PDSWJCL    01.02  93/06/18  93/06/19  9:05    24    17     0  SER09
PDS230I  CUTJCL      01.22  93/06/18  93/06/20  6:53   179   168     0  SER09
PDS117I 7 members counted; cumulative size is 674 records

>----->listgrp
MEM (ADD*
 CUT/
 JCL/)
***************************** BOTTOM OF DATA **********************************
```

LISTV Function

The LV or LISTV command displays disk volume space usage in an ISPF table.

Example

```
LISTV   stor:stor03
```

Syntax

```
LISTV   mask  [RESET] [ADDR|GENERIC]
```

Aliases

```
LV, LISTV
```

Defaults

none – add to the LISTV table

Required

Operands

mask	If ADDR or GENERIC do not follow this parameter, displays all volumes that match the volume mask (mask may contain 1 to 6 characters. An ***** or : can display all volumes.Use a **/** for a pattern specification. Use an ***** for a combination name. Use a : for a range. For example: `TSO/03,TSO*,TSO:TSO80.`
RESET	Clears the LISTV table before adding any new entries.
ADDR	Displays all volumes that match the partial UCB address (name can contain 1 to 3 characters). For example: 0C
GENERIC	Displays all volumes that map to the system UNITNAME (name can contain 1 to 8 characters). For example: SYSDA.
STGGRP	Displays volumes that match the SMS storage group specified.

Remarks

The LISTV ISPF table displays in response to a LISTV command. When you are in a LISTV display, you can delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, etc. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as either a primary command or as a line command.

There are five defined formats for LISTV. You can select which panels you want to use with the SETPANEL command, as shown below. You can switch between these panel formats with RIGHT and LEFT commands. Following this panel is a sample of each of the defined formats.

The Attributes LISTV panel shows basic volume space information and volume name, device address and type, the mount attributes (PRIV, PUB, or STOR), the volume use count and several status indicators for the volume under the header STAT-INDC:

A	The volume is Allocated.
C	A Catalog on the volume is open; the CAXWA chain is searched to check this.
D	DOS contamination; Indexed VTOC is disabled.
S	The volume is marked SHARED in the UCB.
I	An Indexed VTOC is on the volume.
M	The volume is Managed by SMS.
P	A Page data set on the volume is open.
V	The volume has the VSAM ownership bit on.
R	A CRA (Catalog Recovery Area) is on the volume.

The DATA/MSG field is set to *VTOCERR if the VTOC cannot be input or *RESRV if the volume is reserved. The VTOC Size panel shows VTOC information for systems programmers. This includes mount attributes (PRIV, PUB, or STOR), mount status (RESV, PRES, or SYSRes), storage group name (for SMS-managed volumes), the volume use count (note that it is displayed as ******* if it goes over 999), the VTOC CCHH address, VTOC size in tracks, free DSCB count, and free VIR count.

The following primary commands are supported directly for the LISTV function; for information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ADD	Adds data sets to the LISTF table for all volumes in the LISTV table without prompting.	
ALT[ERNAT]	Displays an alternate view of the LISTV table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
DUAL	Switches to a 2-line view of the LISTV table.	
EDITT[BL] ET[BL])	Enters an edit session on LISTV table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax	`F anystring  [ASIS]`
		`   [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
		`   [ANY|VOLUME|ADDRESS]`

LE[FT]	Rotates through alternate views of the LISTV table. PF keys 10 and 22 are normally set to LEFT.	
LISTV	Prompts for LISTV parameters and adds additional volumes to the LISTV table.	
	Syntax	`LISTV [mask] [RESET] [ADDR|GENERIC]`
LISTVT[OC]	Adds data sets to the LISTF table for all volumes in the LISTV table after a prompt.	
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax `LOCATE pvolume|pucb|totalfree|largestfree|attrib`	
NOR[MAL]	Displays the default view of the LISTV table.	
O[PTIONS]	Provides primary command selection for the LISTV function and syntax assistance.	
OUT[PUT]	Outputs the LISTV table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REF[RESH]	Updates information and status for all lines in the table.	
REM[OVE]	Trims the LISTV table based on a string match.	
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]` `              [ANY|VOLUME|ADDRESS]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the LISTV table. PF keys 11 and 23 are normally set to RIGHT.	

SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [VOLUME|ADDRESS|ATTR|TYPE|SPACE|LARGEST]` `     [ASCEND|DESCEND]`
SPA[CE]	Updates volume space statistics for each volume in the LISTV table.	
STATS	Updates volume attributes for each volume in the LISTV table.	
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	`TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.	
X	Clears the LISTV table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the LISTV table; this is equivalent to X ALL	

The following line commands are supported in the LISTV function:

+	Provides an extension panel for command entry. `  *cm ERR- t="Line command,+">`
=	Repeats the previous line command. `  *cm ERR-  t="Line command,=">`
ADD	Adds data sets to the LISTF table for this volume with no prompting.
DIRF	Tailors batch JCL to set the DIRF bit on for this volume.
INDX	Tailors batch JCL to reverse the indexed VTOC status for this volume (an OS/VTOC will be switched to an indexed status and an indexed volume will be switched to an OS/VTOC).
K	Kills and clears all following line commands.
LF	Invokes LISTF to do a VTOC search on this volume after prompting.
LFEX	Invokes LISTF to do a VTOC search on this volume with no prompting.
LIST	Adds data sets to the LISTF table for this volume after prompting.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
O	Provides line command selection and operand syntax assistance.
REFR	Updates volume space and attributes for this volume.
S	Select line command (normally SPAC, it is set by SETSEL).
SPAC	Refreshes space data and statistics for this volume.
STAT	Refreshes statistics for this volume.
TAG	Mark this table entry with *TAG* in the DATA/MSG field.
UT	Select the extended user line command panel.
VMAP	Invokes the VMAP function on this volume.

VTOC	Invokes the TSO VTOC command to display volume contents in the log.
VUSE	Invokes the VUSE subcommand on this volume.
X	Drops the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTV function:

==	= command. Repeats the previous line command for the range of lines.
LL	LOG command. Copies the range of lines into the log.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
XX	X command. Drops the range of table lines.

```
----------------------------- Set PANEL Defaults -----------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
...
 LISTV Displays in order of RIGHT rotation:
 Show Attributes view  ===> NO   (Yes/No) -- Device Mount Total free LARGEST
 Show Free Space view  ===> YES  (Yes/No) -- Device type and free space
 Show Percent use view ===> YES  (Yes/No) -- Device Mount Percent used
 Show VTOC size view   ===> YES  (Yes/No) -- Device Mount VTOC size Location
 Show Custom view      ===> NO   (Yes/No) -- Device Total ....   (customized?)
 Show Double line view ===> YES  (Yes/No) -- Device Total free/Percent used
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment     ===> 50       volumes processed between messages
 VMAP increment        ===> 500      data sets processed between messages
```

```
----------------------- List Volumes - (Attributes) -------- Row 1 to 3 of 3
COMMAND ===> lv trn101                                     SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM= ----------------------------------
   ------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV  DEV MOUNT USE  ---- TOTAL FREE ----   LARGEST  STATUS
---- NAME  -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM DSCBS  CYLS TRKS -INDC-
    LDA001            21B 3390M2 PR  1  1191    4    2  5792  1191    0 A CSI
    SER005            246 3390M2 PR  6   821  535  139  5486   607   15 A  SIM
    TRN101            808 3390M3 ST 11  1587  261   73  9710   932    0 A CSI
 **************************** Bottom of data *****************************


+----------------+
| TOTAL=       1 |
| -------------- |
| ADDS =       1 |
| USED =     51% |
| TOTAL FREE     |
| CYLS =    1587 |
+----------------+
```

```
----------------------- List Volumes - (Free Space) -------- Row 1 to 4 of 4
COMMAND ===>                                                SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=MVS* -----------------------------
 ----------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV     ----- TOTAL FREE -------   ---- LARGEST EXTENT ---
--- NAME  -------- TYPE   CYLS+TRKS=TRACKS->KBYTES  CYLS+TRKS=TRACKS->KBYTES
    OS39H2          3380     73  13  1108   52605     73    9  1104   52415
    OS39R2          3390M3  341  12  5127  290516    341   10  5125  290403
    SCPMV5          3380E   813 129 12324  585118    165    0  2475  117508
    SER002          3380E  1441  17 21632 1027044   1438    0 21570 1024100
```

```
------------------------ List Volumes - (Percent) --------- Row 1 to 3 of 3
COMMAND ===>                                                SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM= --------------------------------
 ----------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV   DEV MOUNT USE SPACE VTOC FREE FREE    LARGEST   STATUS
---- NAME  -------- ADDR TYPE ATTR CNT  USED USED VIRS DSCBS  CYLS TRKS -INDC-
     LDA001          21B 3390M2 PR   1   46%   3%  284 5792  1191    0 A CSI
     SER005          246 3390M2 PR   6   61%   8%  237 5486   607   15 A  SIM
     TRN101          808 3390M3 ST  11   51%   2%  193 9710   932    0 A CSI
***************************** Bottom of data *****************************
```

```
------------------------ List Volumes - (VTOC Size) -------- Row 1 to 3 of 3
COMMAND ===>                                                SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM= --------------------------------
 ----------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV   DEV  -MOUNT-  STORAGE  USE -----VTOC---- FREE  FREE
---- NAME  -------- ADDR TYPE  ATT STAT CLASS    CNT --CCHH-- SIZE DSCBS VIRS
     LDA001          21B 3390M2 PR PRES            1 00010000  120  5792  284
     SER005          246 3390M2 PR PRES SER90      6 00010000  120  5486  237
     TRN101          808 3390M3 ST PRES           11 0000000B  199  9710  193
***************************** Bottom of data *****************************
```

```
-------------------------- List Volumes - (Custom) ---------- Row 1 to 3 of 3
COMMAND ===>                                                SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM= --------------------------------
 ----------------------------------------------------------------------------
 CMD  VOLUME DATA/MSG DEV  DEV MOUNT USE  --TOTAL FREE--    LARGEST    STATUS
 ---- NAME   -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM   CYLS TRKS  -INDC-
                      STORAGE STATUS SPACE                 ------ VTOC -------
                      CLASS     ---- USED  DSCBS VIRS   SIZE USED  --CCHH--
     LDA001           21B 3390M2 PR   1 1191    4    2   1191    0  A CSI
                          PRES   46%  5792  284    120   3%  00010000
     SER005           246 3390M2 PR   6  821  535  139    607   15  A  SIM
                      SER90    PRES   61% 5486  237    120   8%  00010000
     TRN101           808 3390M3 ST  11 1587  261   73    932    0  A CSI
                          PRES   51% 9710  193    199   2%  0000000B
***************************** Bottom of data *****************************
```

```
-------------------------- List Volumes - (Double) ---------- Row 1 to 3 of 3
COMMAND ===>                                            SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM= ----------------------------------
  ----------------------------------------------------------------------------
CMD  VOLUME DATA/MSG DEV  DEV MOUNT USE  --TOTAL FREE--    LARGEST    STATUS
---- NAME   -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM   CYLS TRKS  -INDC-
                     STORAGE STATUS SPACE                  ------ VTOC -------
                     CLASS      ---- USED   DSCBS VIRS   SIZE USED  --CCHH--
     LDA001          21B 3390M2 PR   1 1191    4     2   1191    0  A CSI
                             PRES 46%   5792   284    120    3%  00010000
ut   SER005          246 3390M2 PR   6  821  535   139    607   15  A  SIM
                     SER90   PRES 61%   5486   237    120    8%  00010000
     TRN101          808 3390M3 ST  11 1587  261    73    932    0  A CSI
                             PRES 51%   9710   193    199    2%  0000000B
****************************** Bottom of data ******************************
```

```
---------------------- ut for LV   user line commands ----------------------
OPTION  ===>

Choose one of the following for volume SER005
    DVOL - DVOL command

Dynamic Commands--note:  < = SER005
LONG - TRP(DVOL < LONG)  /* LONG FORMAT DVOL
OVER - TRP(VTOC < LIM(PCT LT 50) OR1(UNUSED GT 30))
____ -
```

LLA Subcommand

The LLA subcommand selectively refreshes member directory entries in LLA-managed data sets. If a member is not present in the data set, the LLA subcommand removes that member from LLA management.

Example

```
LLA   isr*
```

Syntax

```
LLA   memgroup [ALIAS|NOALIAS]
                [FORCE       ]
                [SYNC        ]
```

Aliases

LLA

Defaults

memgroup, ALIAS

Required

Operands

memgroup	The name of the modules for which the LLA entry is to be refreshed or removed. Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names".
ALIAS	Any associated members (aliases, apparent aliases, and associated main members) are also to be refreshed in the LLA directory with LLACOPY.
NOALIAS	Only the named members are to be refreshed or removed from the LLA directory with LLACOPY.
FORCE	The LLACOPY is to be performed even if StarTool FDM indicates the data set is not LLA-managed. For example, if PARMLIB member CSVLLAxx indicates GET_LIB_ENQ(NO). If this parameter is used and the data set is not managed, a BLDL is performed by the operating system instead of an LLACOPY.
SYNC	Members that are not synchronized with their directory entries are to be selectively refreshed. SYNC can update only changed members in a library thereby saving resources required to needlessly stage members. SYNC processing is only available for LLA-managed, LINKLIST data sets because a system BLDL is used to determine a member's LLA status. Therefore, this routine cannot determine the LLA status of any members that are also in a TASKLIB data set. **NOTE** SYNC processing compares directory entries provided by LLA against directory entries from disk to determine if members are synchronized. This type of check cannot detect a zap or update in place.

Remarks

Internally, the LLA subcommand issues an LLACOPY macro to refresh LLA entries. LLACOPY does not refresh LLA if:

- The current data set is not managed by LLA

- LLA is not active; there is no LLA directory to update

- System level does not support LLA

When you first enter an LLA managed library with StarTool FDM or you request a USAGE subcommand, you should get the feedback message:

```
PDS189I This data set is managed by LLA”
```

If you attempt to use the LLA subcommand and LLA is not active or the data set is not managed by LLA, you should get the error message:

```
“PDS727E This data set is not managed by LLA”
```

If the operating system level does not support LLA, LLACOPY returns codes associated with BLDL. That is, a member that is present gets a message saying the LLA entry was refreshed. A member that is not present gets a message saying the LLA entry was removed.

Issue an 'LLA REFRESH' command to resolve these error messages.

LLA and Deleted Members

Use the LLA subcommand as a line command in a MEMLIST table to manage deleted members.

If you are planning to *rename* members of an LLA controlled data set, set "Remove renamed members" in SETALL for MEMLIST to NO. Then, after a member is renamed with the REN line command, enter an 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 SETALL for MEMLIST to NO. Then, after a member is deleted with the DEL line command, enter an LLA line command on the deleted member name to inform LLA that the member has been deleted from the data set. After the LLA subcommand finishes, you can remove the deleted member with an X line command.

```
------------------------- ISPMODE Session# 2 Log # 1 -- ROW 195 to 197 of 197
COMMAND ===>                                             SCROLL ===> CSR
- DSN=SYS1.V310.SYSLINK,VOL=SER=SYSPDS  MEM=STARTOOL  -----------------------
>----->lla (startool,oldmod)
PDS073I STARTOOL has been refreshed
PDS073I PDSTOOLS has been refreshed
PDS074I OLDMOD has been removed
PDS073I PDSE has been refreshed
PDS073I PDS has been refreshed
****************************** BOTTOM OF DATA ******************************
```

LOG Function

The LOG command displays the current or one of nine session logs. LOG table 1 is the default. It is the table displayed when dialog mode is first started.

Example
LOG

Syntax
LOG   [num] [RESET] [NODISPL]

Aliases

`LO, LOG, 8`

Defaults

1 initially; otherwise, the current log table.

Required

Operands

num	Identifies the log table desired (1 through 9).
RESET	Empties the selected log table and initializes it with the PDS300I default message.
NODISPL	Sets the LOG function to use the selected numbered session log; however, it does not cause a switch from the current function. RESET and NODISPL are often used together to clear a log table without transiting to the nullified log table.

Remarks

The ISPMODE, ISPXEQ, or MEMLIST subcommand is used to first enter dialog mode. After you are in dialog mode, use the LOG command to display any one of nine log tables.

The log is displayed in response to a LOG command or to display the output from a subcommand. When you are in the log table, you can delete a part of the log, find data in the log, print a part of the log, store a part of the log in a data set, etc. For assistance with the different options, use the HELP command, CUA functions, or enter an O command as a primary command.

In the following, LOG session processing is described in more detail. The PUTLINE and PUTGET routines are monitored for data.

- When the PUTLINE routine is entered, the dialog adds another log table entry.

- If an output checkpoint is reached for a subcommand (by default, 250 output lines as set by SETALL for the LOG), ISPMODE positions the log table display to the beginning of the output of this subcommand and allows you to continue or terminate. You must respond to this screen before you can enter any StarTool FDM subcommands. Some useful commands in this situation are:

C	Continue with the output from the subcommand.
END	Terminate the subcommand (using a simulated attention).
F	Find data in the log.

- By default, CONTROL PROMPT is enabled. This enables prompting in several situations with StarTool FDM messages PDS390A through PDS399A at critical decision points in StarTool FDM processing. StarTool FDM operates in *reply required* mode until you

respond to these messages. You cannot enter other StarTool FDM subcommands. Some useful commands in this situation are:

CAN	On a PDS390A prompt from RESTORE, enter CAN to terminate RESTORE processing.
F	Find data in the log.
NO	Indicates that StarTool FDM *not* complete the current action.
YES	Indicates that StarTool FDM continue with the current action.

- When the PUTGET routine is entered, the dialog positions the log display to the beginning of the output of the just completed subcommand. Enter any type of subcommand or command at this time.

The following primary commands are supported in the LOG function. For documentation on ISPMODE commands available anywhere, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

ALT[ERNAT]	Displays an alternate view of the log.
BAT[CHJCL]	Builds batch JCL that includes all StarTool FDM statements in the LOG table.
CAN	After a reply required for RESTORE, specifies that StarTool FDM terminate RESTORE processing. A YES response allows the restoration of the current deleted member and a NO response skips the current deleted member for RESTORE processing.
COLS	Provides a column ruler in the log for determining column numbers.
CONT[INUE]	After a checkpoint, specifies that StarTool FDM continue the current interrupted process until the next checkpoint interval as specified by SETALL for the LOG. Use END to terminate the process.
DUA[L]	Displays a double line view of the log table.
EDITL[OG] (EL[OG])	Enters an edit session on the output from the last subcommand.
EDITT[BL] (ET[BL])	Enters an edit session on data from the log.
F	Finds a string. Resets the display start location and positions the cursor.
	Syntax `F anystring [nn mm] [ASIS]` `[FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	(where *nn* and *mm* are column numbers)
LE[FT]	Rotates through alternate views of the log table. PF keys 10 and 22 are normally set to LEFT.
MERGE	Adds members to the MEMLIST table from a saved MEMLIST table.
	Syntax `MERGE  name  [RESET] [NOSTATS]`
	(where name is 1 to 6 alphanumeric characters)

NO	After a required reply, specifies that StarTool FDM not complete the current action. A YES response allows the process to continue. For a RESTORE subcommand, enter CAN to terminate RESTORE processing.
NOR[MAL]	Displays the default view of the log table.
O[PTIONS]	Provides primary command selection for the log function and operand syntax assistance.
OUT[PUT]	Outputs the log table to print or a data set.
	Syntax `OUTPUT [=c	F(ddname)]`
RCH[ANGE]	Captures the command or data set name under the cursor in the log for modification and/or reuse. PF keys 6 and 18 are normally set to RCHANGE.
RF[IND]	Finds a string (repeat find). Resets the display start location and positions the cursor over the string. PF keys 5 and 17 are normally set to RFIND.
RI[GHT]	Rotates through alternate views of the log table. PF keys 11 and 23 are normally set to RIGHT.
S[ELECT]	BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.
	Syntax `SELECT member`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. The log UT panel also allows dynamic primary commands with which you specify command names and their corresponding actions.
X	Clears the log table relative to the cursor position.
	Syntax `X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the log table. This is equivalent to X ALL
Y[ES]	After a required reply, specifies that StarTool FDM continue the current action. A NO response stops the current action. For a RESTORE subcommand, enter CAN to terminate RESTORE processing.

General LOG Notes

1 Only the first 159 characters of an output line are retained in the log. This can be a consideration in LIST, FIND, or REPLACE with long lines and with list formats NUM, SNUM, or NONUM.

2 Echoed subcommand lines are identified with >------> before the subcommand. Continuation lines are identified by a **-** character in the first log position.

3 The F command is a log search command and not the StarTool FDM FIND subcommand. If you want the FIND subcommand, enter FI, FIN, or FIND.

4 Subcommand names PRINT, HELP, and TSO are intercepted by ISPF. To use the StarTool FDM subcommands with the same names, use an abbreviated name such as PRI, HE, or TS.

5 Since output is displayed only at a checkpoint or at the completion of a subcommand, output from a subcommand cannot be monitored as it is produced. Therefore, if StarTool FDM is in a CPU loop, the screen locks until it receives an attention. At that time, any outputs generated display and an appropriate response is given. For more information, see Appendix D, "Attention Processing".

6 If a StarTool FDM subcommand is too long for the 48 character command field, enter the first 47 or less characters followed by a **+** to indicate that more data is required. A continuation panel with more room for operand entry displays.

7 The END command does not work as in StarTool FDM line mode. END terminates the current function. Multiple END commands are normally required to terminate the StarTool FDM command. Use the QUIT subcommand to terminate StarTool FDM.

```
------------------------ ISPMODE Session# 1 Log# 1 ------- Row 1 to 27 of 59
COMMAND ===> batchjcl                                    SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=STARXYZ*  --------------------------
>------>dup starxyz* lib.cntls
PDS852E STARXYZ7 already exists
PDS051I STARXYZE was copied; input=27; output=27
PDS051I STARXYZJ was copied; input=15; output=15
PDS051I STARXYZM was copied; input=26; output=26


PDS054I Totals -- Members=3; Input=44; Characters=5,440


>------>change lib.cntls
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80   32720    1X     10       5     10 TRK      6


PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001   MEM=STARXYZ*
>------>compress
IEB1135I IEBCOPY  FMID HDZ11C0 SERVICE LEVEL UW30359  DATED 19960711 DFSMS 1.3
IEB1035I WSER07    ISPFPROC ISPFPRO2 07:45:26 WED 23 APR 1997 PARM=''
IEB1064I   STANDARD DD NAMES- SYSIN    SYSPRINT SYSUT1   SYSUT2   SYSUT3  IEB1065I
    OVERRIDING DD NAMES- SYS00017 SYS00018 SYS00016 SYS00016 SYS00019 IEB1057I VL
    GETMAIN REQUESTED 250K TO 1M BYTES.  OBTAINED 1M.
ISPFPRO2 COPY      INDD=SYS00016,OUTDD=SYS00016     GENERATED STATEMENT
IEB1058I ALLOCATED 2 CONTIGUOUS BUFFERS EACH 94976 BYTES.  WORK AREA HAS 826K
IEB1018I COMPRESSING  PDS  OUTDD=SYS00016 VOL=SER001 DSN=WSER07.LIB.CNTLS
IEB1106I CONTROL TABLE IS 210 BYTES LONG.  WORK AREA HAS 825K BYTES AVAILABLE.
IEB1103I FIRST GAS BUBBLE IS BEFORE MEMBER CARDIN    AT TTR=X'00000C'
IEB1097I FOLLOWING MEMBER(S) MOVED IN DATA SET REFERENCED BY SYS00016
IEB1098I 21 OF 21 MEMBERS MOVED IN DATA SET REFERENCED BY SYS00016
IEB144I THERE ARE 8 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYS00016
IEB149I THERE ARE 6 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB1056I RELEASED 1016K ADDITIONAL BYTES.
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE


PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80   32720    1X     10       8     10 TRK      6
```

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
_____
 EDIT       WSER07.SPFTEMP1.CNTL                         Columns 00001 00072
 Command ===>                                              Scroll ===> CSR
 ****** **************************** Top of Data ****************************
 000001 //WSER07A JOB (X170,374),'SHOWS PROTOTYPING',CLASS=A,
 000002 //   MSGCLASS=X,TIME=(0,30),NOTIFY=WSER07
 000003 //PDS05021   EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=5M
 000004 //SYSPRINT DD SYSOUT=*
 000005 //SYSTSPRT  DD  SYSOUT=*
 000006 //SYSTSIN   DD  *
 000007 dup starxyz* lib.cntls
 000010 change lib.cntls
 000011 compress
 ****** ************************* Bottom of Data ***************************
```

```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
-----------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,009 OF 1,009
COMMAND ===> ut                                            SCROLL ===> CSR
 - DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=ANY/ ---------------------------- -
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SER007 IS 0
>------>DEFINE CLUSTER ( NAME(SER07.VSAM.T.CLUSTER) INDEXED VOL(SER007) RECORD
-SIZE(200 9000) SPANNED KEYS(8 0)  ERASE NORECATAL
-OG NOREPLICATE NOREUSE NOIMBED SPEED WRITECHECK BUFFERSPACE(40000) FREESPACE(
-33 15) SHAREOPTIONS(2 3)) DATA ( TRACKS(1 1
-) CISZ(4096) ) INDEX ( TRACKS(1 1) CISZ(4096) )
IDC0508I DATA ALLOCATION STATUS FOR VOLUME SER007 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SER007 IS 0
IDC0512I NAME GENERATED-(D) SER07.VSAM.TS.DATA
IDC0512I NAME GENERATED-(I) SER07.VSAM.TS.INDEX
>----->c lib.test
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB      80    9040    1X    47      10     40 TRK     24

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=ANY/
****************************** BOTTOM OF DATA ******************************
```

```
----------------------- UT for a PDS   User Commands ----------------------
OPTION  ===>

Choose one of the following:
       Z - ZAP command processor
    MINE - Members last saved by SER07
   MONTH - Members updated in the last month

Dynamic Commands--note:  < = ,
                      / = 'SER07.LIB.TEST'
 SETSYS1_ - WORKPAD 12 SETUP  /* SETUP MY SYS1.* SESSION
 LISTHIST - TRP(LISTD / HISTORY)
 LOAD____ - GO LIB.LOAD
 PROCLIB_ - GO 'SYS1.PROCLIB' VOL(SYSRES)
 _____ -
```

MAP Subcommand

The MAP subcommand lists the CSECT structure of a load module. It can also modify the addressing mode and residence mode of specific module CSECTS. The map it produces is similar to the output produced by the MAP option of the linkage editor.

Example

```
MAP    mema:memb
```

Syntax

```
MAP  memgroup [FULL|SHORT|ENTRY|RELINK|JCL|LCT                        ]
              [EXTERN|WKEXTERN|PSEUDOREG|LABELREG|COMMON|PRIVATE|CSECT]
              [OFFSET(hexoff)                                         ]
              [NOSYSTEM                                               ]
              [MODULE({* | Fullm | Partm*)}                          ]
              [REPLACE                                                ]
              [AMODE24|AMODE31|AMODE64 AMODEANY  RMODE24|RMODEANY     ]
               [MEMBERS | MEMLIST | ML | NEWML | SUBLIST             ]
```

Aliases

```
MA, MAPD
```

Defaults

memgroup, FULL

Required

Operands

memgroup	Identifies the load members for which you want map information. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
FULL	Provides a full CSECT map. This option also provides the name of the corresponding main module for alias members unless CONTROL NOALIASINFO is in effect.
SHORT	Provides a CSECT map with entry names within CSECTS, missing weak external references, and alias member names omitted.
ENTRY	Displays only the ENTRY point symbol.
RELINK	Generates JCL and linkage editor control statements for use in relinking this module.

JCL	Generates JCL and linkage editor control statements for use in relinking this module.
LCT	Generates commented linkage attributes and linkage editor control statements for use in managing this module.
OFFSET(hexoff)	Specifies a 1- to 8-character hexadecimal offset for the origin of this load module.
EXTERN	Load modules with missing external symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
WKEXTERN	Load modules with missing weak symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
PSEUDOREG	Load modules with PSEUDO register symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
LABELREF	Load modules with ENTRY symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
COMMON	Load modules with COMMON area symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
PRIVATE	Load modules with PRIVATE area symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
CSECT	Load modules with CSECT symbols are wanted. The MODULE keyword can also be specified to search for specific symbols.
MODULE(nm)	Specifies a 1- to 8-byte partial external name that limits CSECT and ENTRY names for map reporting. The MODULE operand has several valid forms: ■ **MODULE(*) –** uses the previous name entered on any MODULE keyword ■ **MODULE(Fullm) –** reports only on CSECT or ENTRY name FULLM ■ **MODULE(Partm*) –** reports only on CSECT or ENTRY name PARTM...
REPLACE	Specifies with **RELINK**, **LCT**, or **JCL** to indicate all CSECTS are to be listed with binder REPLACE statements before the binder INCLUDE statement. **REPLACE** can also be used with the **MODULE** keyword to select module names for replacement.
AMODE24	Sets the addressing mode of CSECTS identified by MODULE to 24.
AMODE31	Sets the addressing mode of CSECTS identified by MODULE to 31.
AMODE64	Sets the addressing mode of CSECTS identified by MODULE to 64.
RMODE24	Sets the residence mode of CSECTS identified by MODULE to 24.
RMODE64	Sets the addressing mode of CSECTS identified by MODULE to 64.
AMODEANY	Sets the addressing mode of CSECTS identified by MODULE to ANY.
RMODEANY	Sets the addressing mode of CSECTS identified by MODULE to ANY.

NOSYSTEM	System modules are to be filtered out before reporting MAP information as is done for HISTORY GENERATE. System or compiler routines begin with DFH, DFS, DSN, IBM, IEY, IGY, ILB, ISP, or PLI.
MEMBERS	Displays the names of members that satisfy the MAP subcommand without changing the current member group.
MEMLIST	Same as ML. Any member displayed by the MAP subcommand is selected for MEMLIST display. The MODULE(name) keyword searches for members to display. If no members are selected, a null sublist is the result.
ML	Same as MEMLIST. Any member displayed by the MAP subcommand is selected for MEMLIST display. The MODULE(name) keyword searches for members to display. If no members are selected, a null sublist is the result.
NEWML	Same as MEMLIST and ML except that the current MEMLIST is reset.
SUBLIST	Any member displayed by the MAP subcommand is selected for inclusion in a new sublist. The MODULE(name) keyword searches members to display. If no members are selected, a null sublist is the result.

```
 FUNCTIONS   CONTROL   DSN CMDS   MEM CMDS A-M   MEM CMDS N-Z   DEFAULTS   FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 115 OF 127
COMMAND ===> map zap$                                      SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAP$  --------------------------
** MAP     ZAP$
ZAP       00000000  00003F06
PCL       00003F08  000000E5
CSOUT     00003FF0  00000428
ZAPHELP   00004418  00001B1C
ASMGASM   00005F38  000007E4
PDS103I Entry point at 00000000  --  ZAP
PDS104I Module length  00006720  --    27K
PDS066I Member is an alias for: ZAPMAIN
```

```
----------------------- ISPMODE Session# 1 Log# 1 --- Row 264 to 280 of 280
COMMAND ===>                                             SCROLL ===> CSR
- DSN=IBMUSER.LIB.LOAD,VOL=SER=SER001  MEM=DSAT ---------------------------
>------>map dsat

** MAP       DSAT
DSAT     00000000 00001CC8  RMODE 24  AMODE 64
DSATPDS  00001CC8 00000498  RMODE 24  AMODE 64
DSATPCL  00002160 000003F8  RMODE 24  AMODE 64

PDS103I Entry point at 00000000  --  DSAT
PDS104I Module length  00002558  --    10K
>------>map dsat mod(dsatp*) amode31 amode64 rmodeany

** MAP       DSAT
DSATPDS  00001CC8 00000498  RMODE ANY AMODE 31
DSATPCL  00002160 000003F8  RMODE ANY AMODE 31

PDS103I Entry point at 00000000
PDS104I Module length  00002558  --    10K
***************************** Bottom of data *****************************
```

```
-------------------------- ISPMODE Session# 1 Log# 1 --- Row 464 to 480 of 480
 COMMAND ===>                                            SCROLL ===> CSR
 - DSN=IBMUSER.LIB.LOAD,VOL=SER=SER001  MEM=DSAT -------------------------------
>------>map dsat64 relink replace

** MAP       DSAT64
//LKED   EXEC  PGM=IEWL,
//       PARM='NCAL,MAP,LIST,LET,RENT,REUS,REFR'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=WSER07.LINK.LOAD
//SYSLMOD  DD  DISP=SHR,DSN=WSER07.LINK.LOAD
//SYSLIN   DD  *
 REPLACE DSAT,DSATPDS,DSATPCL
 INCLUDE SYSLIB(DSAT64)
 ORDER   DSAT64,DSATPDS,DSATPCL
 MODE    RMODE(ANY),AMODE(64)
 SETCODE AC(1)
 ENTRY   DSAT64
 NAME    DSAT64(R)
```

```
-------------------------- ISPMODE Session# 1 Log# 1 --- Row 334 to 350 of 350
 COMMAND ===>                                               SCROLL ===> CSR
 - DSN=IBMUSER.LIB.LOAD,VOL=SER=SER001  MEM=DSAT  -----------------------------
>------>map dsat64 relink replace module(dsatp*)

** MAP      DSAT64
//LKED   EXEC  PGM=IEWL,
//        PARM='NCAL,MAP,LIST,LET,RENT,REUS,REFR'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=WSER07.LINK.LOAD
//SYSLMOD  DD  DISP=SHR,DSN=WSER07.LINK.LOAD
//SYSLIN   DD  *
 REPLACE DSATPDS,DSATPCL
 INCLUDE SYSLIB(DSAT64)
 ORDER   DSAT64,DSATPDS,DSATPCL
 MODE    RMODE(ANY),AMODE(64)
 SETCODE AC(1)
 ENTRY   DSAT64
 NAME    DSAT64(R)
```

```
-------------------------- ISPMODE Session Display ---------- ROW 635 OF 649
COMMAND ===> map zap$ relink                                SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAP$  ---------------------------
** MAP      ZAP$
//LKED   EXEC  PGM=IEWL,
//        PARM='NCAL,MAP,LIST,LET'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=C911407.LINK.LOAD
//SYSLMOD  DD  DISP=SHR,DSN=C911407.LINK.LOAD
//SYSLIN   DD  *
 INCLUDE SYSLIB(ZAPMAIN)
 ORDER   ZAP,PCL,CSOUT,ZAPHELP,ASMGASM,OACNOW
 ENTRY   OACNOW
 ALIAS   ZAP,ZAP$,ZAPHELP
 NAME    ZAPMAIN(R)
```

MASK Function

The MASK command is an extension to the LISTC and LISTF commands. It allows you to enter a data set mask or a volume mask and to then optionally execute the LISTC or LISTF function.

Example

```
MASK   ser07.lib.*
```

Syntax

```
MASK mask [n|VOLUME|VOLUME(vmask)   ]
          [CLEAR|NOCLEAR             ]
          [UPDATE|LC|LISTC|LF|LISTF]
          [RESET                     ]
```

```
[NOFILTER                    ]
[PROMPT                      ]
```

Aliases

MAS, MASK

Defaults

1, CLEAR, LISTC

Required

mask

Operands

mask	Identifies the mask that is to be applied to LISTC or LISTF.
	For the LISTF volume mask, user a **/** for a pattern specification; use an ***** for a combination name, and a **:** for a range. For example: TSO/03, TSO*, and TSO:TSO80.
	If the mask is for a data set name, each level can contain an ***** before or after any characters in that level. For example: SYS1.NU*.*.*XY. The first level can contain an ***** after a beginning character string.
n|VOLUME	Identifies the position that this mask is to occupy (1 - 3 for a data set mask) or VOLUME for a volume mask.
	The VOLUME keyword can contain a volume mask. If you do this, the first mask is considered a data set mask; otherwise, the first mask is considered a volume mask.
CLEAR	Clears out the other mask entries.
NOCLEAR	Leaves the other mask entries.
UPDATE	Updates the mask only.
LC	Executes LISTC with this mask.
LISTC	Executes LISTC with this mask.
LF	Executes LISTF with this mask.
LISTF	Executes LISTF with this mask.
RESET	Clears the current LISTC/LISTF table before executing LISTC or LISTF.
NOFILTER	Ignores all FILTER options and resets all of the FILTER options to their defaults.
PROMPT	Displays MASK prompt panel after updating with string.

Remarks

MASK provides a way to update LISTC/LISTF parameters without prompting. You can use it from a WORKPAD to build and execute a complicated LISTF table. The following MASK

command builds a data set table from a catalog search of data sets using the single data set mask, USERID.*.CNTL:

```
MASK USERID.*.CNTL
```

The following MASK command builds a data set table from a VTOC search of all PUBxxx volumes using the single data set mask, USERID.*.CNTL:

```
MASK USERID.*.CNTL VOLUME(PUB*) LISTF
```

The following MASK commands build up 3 data set masks. The first MASK command clears all previous masks and the next one adds an additional data set mask. The last MASK command adds a data set mask, a volume mask, and starts the VTOC search to build the LISTF table.

```
MASK SYS1.* 1 UPDATE CLEAR
MASK SYS2.* 2 UPDATE NOCLEAR
MASK SYS3.* 3 VOLUME(RES*) NOCLEAR LISTF
```

The above example could also have been specified as:

```
MASK SYS1.* 1 UPDATE CLEAR
MASK SYS2.* 2 UPDATE NOCLEAR
MASK SYS3.* 3 UPDATE NOCLEAR
MASK RES* VOLUME     NOCLEAR LISTF
```

The following MASK command continues a data set search to additional volumes starting with SYS:

```
MASK SYS* VOLUME     NOCLEAR LISTF
```

MEMBERS Subcommand

The MEMBERS subcommand lists members that are in the current member group. It is often used to check member names in a given group before invoking another subcommand that processes member groups.

Example
```
MEMBERS    mema:memb
```

Syntax
```
MEMBERS    memgroup [COUNT | NOCOUNT| ONLYCOUNT   NOBLDL]
```

Aliases
```
ME, MEM, MEMB, MEMBE, MEMBER, MEMBERS
```

Defaults

memgroup, COUNT

Required

Operands

memgroup	Identifies the members whose names are to be displayed. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
COUNT	Issues a summary message to display the number of members in the current group.
NOCOUNT	Does not issue a summary count message.
ONLYCOUNT	Issues only a message indicating the number of members in the current group.
NOBLDL	The existence of each member in the current group need not be verified with a BLDL macro. This parameter is intended for high-performance applications in conjunction with the SUBLIST subcommand.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 150 OF 155
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=(ASID  -------------------------
>----->if : rmode31 then(sublist)
>----->members
PDS165I Members are: ASID, ASIDZN, JOB0LIST, MAPXA, PDSPGM, SORT, TSDYNLXA,
        WHAT, WHATO, X$SPACE, XVOLS

PDS193I This group contains 11 members
***************************** BOTTOM OF DATA  *****************************
```

MEMLIST Function

The MEMLIST subcommand builds a scrollable member list display on which StarTool FDM subcommands can be entered as line commands to process individual members. Enter ISPF commands, StarTool FDM subcommands, and primary commands to the right of COMMAND==>. For additional information, see Appendix A, "Formatting Member Names".

Example

```
MEMLIST    mema:memb
```

Syntax

```
MEMLIST memgroup
       [SINCE|BEFORE                                        ]
```

```
        [TODAY|YESTERDAY|WEEK|CURRENT|BIWEEK|
            MONTH|QUARTER|HALFYEAR|YEAR|BIYEAR|
            LAST(numdays)|DATE(yyyy/mm/dd)      ]
        [CHANGED(yyyy/mm/dd:yyyy/mm/dd)         ]
        [CREATED(yyyy/mm/dd:yyyy/mm/dd)         ]
        [ABOVE(Count1)                          ]
        [ALIAS|NOALIAS                          ]
        [AMODE24|AMODE31|AMODE64|AMODEANY|NOAMODE24|
            NOAMODE31|NOAMODE64|NOAMODEANY      ] (load only)
        [AUTH|NOAUTH                            ] (load only)
        [BELOW(Count2)                          ]
        [DC|NODC                                ] (load only)
        [EDIT|NOEDIT                            ] (load only)
        [EXEC|NOEXEC                            ] (load only)
        [FLEVEL|NOFLEVEL                        ] (load only)
        [HASALIAS|NOHASALIAS                    ]
        [ID(Puid)|NOID|NOTID(Puid)              ]
        [LKED(Partl)                            ] (load only)
        [LOADONLY|NOLOADONLY                    ] (load only)
        [MODULE(* | Fullm | Partm*)            ] (load only)
        [NULL|NONULL                            ]
        [ORPHAN|NOORPHAN                        ]
        [OVERLAY|NOOVERLAY                      ] (load only)
        [PAGE|NOPAGE                            ] (load only)
        [REFR|NOREFR                            ] (load only)
        [RENT|NORENT                            ] (load only)
        [REUS|NOREUS                            ] (load only)
        [RMODE24|RMODEANY|NORMODE24|NORMODEANY] (load only)
        [SCTR|NOSCTR                            ] (load only)
        [SSI(hxdata)|SSI|NOSSI|PARTSSI(hxdata)]
        [SYSMOD(Partu) | USERDATA(Partu)        ] (load only)
        [TEST|NOTEST                            ] (load only)
        [TRANS(Partt)                           ] (load only)
        [TTR(Lttr:Httr)                         ]
        [USERDATA(Partu) | SYSMOD(Partu)        ] (load only)
        [USERID(Puid)|NOUSERID|NOTUSERID(Puid)]
        [VSLKED|NOVSLKED                        ] (load only)
        [ZAP(Partz)                             ] (load only)
        [RESET|NORESET                          ]
        [LKEDDATE|NOLKEDDATE                    ] (load only)
```

Aliases

ML, MEML, MEMLI, MEMLIS, MEMLIST

Defaults

memgroup, NORESET, LKEDDATE, SINCE

Required

Operands

memgroup	Identifies the members that are to be added to the member list.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names"**.**
SINCE	Uses dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
FROM	Uses dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
BEFORE	Uses dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
TO	Uses dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.
TODAY	A date, equivalent to LAST(0).
YESTERDAY	A date, equivalent to LAST(1).
WEEK	A date, equivalent to LAST(7).
CURRENT	A date, equivalent to LAST(10).
BIWEEK	A date, equivalent to LAST(14).
MONTH	A date, equivalent to LAST(30).
QUARTER	A date, equivalent to LAST(120).
HALFYEAR	A date, equivalent to LAST(183).
YEAR	A date, equivalent to LAST(365).
BIYEAR	A date, equivalent to LAST(730).
LAST(numdays)	A date indicating the number of days before today.
DATE(cdate)	A date indicating the actual date; enter it in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).
CHANGED(date:	Searches for members modified by an ISPF editor or a ZAP program between the dates specified. If you enter only a single date, only that date is checked, but normally a date range like CHANGED(yyyy/mm/dd:yyyy/mm/dd) is entered.
	Only members with ISPF statistics or members created by a linkage editor are supported.

CREATED(date:	Searches for members created with ISPF statistics or by the linkage editor between the dates specified. If you enter only a single date, only that date is checked, but normally a date range like CREATED(yyyy/mm/dd:yyyy/mm/dd) is entered.
ABOVE(Count1)	Selects members with more than the number of lines (amount of storage for load modules) defined by the number Count1.
ALIAS	Selects alias members.
NOALIAS	Selects main members.
AMODE24	Selects modules with addressing mode 24.
NOAMODE24	Selects modules with addressing mode 31 or ANY.
AMODE31	Selects modules with addressing mode 31.
NOAMODE31	Selects modules with addressing mode 24 or ANY.
AMODE64	Selects modules with addressing mode 64.
NOAMODE64	Selects modules with addressing mode 64 or ANY.
AMODEANY	Selects modules with addressing mode ANY.
NOAMODEANY	Selects modules with addressing mode 24, 31 or 64.
AUTH	Selects APF-authorized modules.
NOAUTH	Selects non-authorized modules.
BELOW(Count2)	Selects members with less than the number of lines (amount of storage for load modules) defined by the number Count2.
DC	Selects downward-compatible modules.
NODC	Selects no downward-compatible modules.
EDIT	Selects modules that can be reprocessed by the linkage editor.
NOEDIT	Selects modules that cannot be reprocessed by the linkage editor.
EXEC	Selects EXECUTABLE modules.
NOEXEC	Selects non-EXECUTABLE modules.
FLEVEL	Selects modules processed by the F-level linkage editor.
NOFLEVEL	Selects modules not processed by the F-level linkage editor.
HASALIAS	Selects main members that have one or more aliases.
NOHASALIAS	Selects alias members or main members without any aliases.
ID(Puid)	Selects members with ISPF statistics and user IDs matching the partial name Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. ID is an alias of USERID.
NOID	Selects members without ISPF statistics. NOID is an alias of NOUSERID.
NOTID(Puid)	Same as NOTUSERID. Selects members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters.

LKED(Partl)	Selects modules identified as linked by the linkage editor defined by the partial name Partl. Enter Partl as a partial linkage editor IDR name with 1 to 10 characters.
LOADONLY	Selects modules marked for LOAD ONLY.
NOLOADONLY	Selects modules not marked for LOAD ONLY.
MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT names for attribute searches. If you enter this parameter with any TRANS, SYSMOD/ USERDATA, or ZAP keywords, only those CSECTs that satisfy the MODULE keyword are checked for the other keyword conditions. The MODULE operand has several valid forms: ■ MODULE(*) – uses the last name entered on a MODULE keyword ■ MODULE(Fullm) – selects any module containing a CSECT or ENTRY called FULLM ■ MODULE(Partm*) – selects any module containing a CSECT or ENTRY called PARTM...
ORPHAN	Selects alias members that have no associated main member.
NOORPHAN	Selects main members or alias members that have an associated main member.
OVERLAY	Selects OVERLAY modules.
NOOVERLAY	Selects non-OVERLAY modules.
PAGE	Selects modules marked for loading on a page boundary.
NOPAGE	Selects modules not marked for loading on a page boundary.
REFR	Selects refreshable modules.
NOREFR	Selects non-refreshable modules.
RENT	Selects reentrant modules.
NORENT	Selects non-reentrant modules.
REUS	Selects reusable modules.
NOREUS	Selects non-reusable modules.
RMODE24	Selects modules with residence mode 24.
NORMODE24	Selects modules with residence mode ANY.
RMODEANY	Selects modules with residence mode ANY.
NORMODEANY	Selects modules with residence mode 24.
SCTR	Selects scatter-loaded modules.
NOSCTR	Selects no scatter-loaded modules.
SSI(hxdata)	Selects members with matching SSI data. This is implemented as a generic search matching SSI characters from left to right for the number of characters entered.
SSI	Selects members with SSI data.
NOSSI	Selects members without SSI data.

PARTSSI(hxdata)	Selects members with matching SSI data. This is implemented as a pattern search matching SSI digits as a string anywhere in the SSI field of the member.
SYSMOD(Partu)	Selects modules with user IDR data that matches the partial name Partu. Enter Partu as partial IDR data with 1 to 8 characters. If MODULE(...) is also entered, the SYSMOD operand applies only to CSECTs whose names satisfy the MODULE argument.
TEST	Selects modules with the TEST attribute.
NOTEST	Selects modules without the TEST attribute.
TRANS(Partt)	Selects modules with CSECTs identified as having been assembled or compiled by the translator defined by the partial name Partt. Enter Partt as a partial translator IDR name with 1 to 10 characters. If MODULE(...) is also entered, the TRANS operand applies only to CSECTs whose names satisfy the MODULE argument.
TTR(Lttr:Httr)	Selects members whose start address is in the specified TTR range. Lttr defaults to 0 and can be entered as a hexadecimal TTR value from 0 through FFFFFF. Httr defaults to FFFFFF and can be entered as a hexadecimal TTR value from 0 through FFFFFF.
USERDATA(Partu)	Selects modules with user IDR data that matches the partial name Partu. Enter Partu as partial IDR data with 1 to 8 characters. If MODULE(...) is also entered, the USERDATA operand applies only to CSECTs whose names satisfy the MODULE argument.
USERID(Puid)	Selects members with ISPF statistics and user IDs matching the partial name Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. USERID is an alias of ID.
NOUSERID	Selects members without ISPF statistics. NOUSERID is an alias of NOID.
NOTUSERID(Puid)	Selects members without ISPF statistics or members with ISPF statistics and user IDs that do not match the partial name Puid. Enter Puid as a partial TSO user ID with 1 to 7 characters. NOTUSERID is an alias of NOTID.
VSLKED	Selects modules linked by the system linkage editor.
NOVSLKED	Selects modules not linked by the system linkage editor.
ZAP(Partz)	Selects modules with zap IDR data that matches the partial name Partz. Enter Partz as partial zap IDR data with 1 to 8 characters. If MODULE(...) is also entered, the ZAP operand applies only to CSECTs whose names satisfy the MODULE argument.
RESET	Drops any current members from the member list and adds the member group named with this subcommand.
NORESET	Merges the current member group with the existing member list.
LKEDDATE	Provides linkage edit date info for load members (IDR data must be read).
NOLKEDDATE	Does not provide that linkage edit date information for load members.

Remarks

MEMLIST line commands and their outputs are maintained in the log.

If a StarTool FDM subcommand is too long for the 48-character input field, enter the first 47 or less characters followed by a **+** to indicate more data is required. A continuation panel with more room for operand entry displays.

The MEMLIST ISPF table displays in response to a MEMLIST subcommand issued in StarTool FDM line mode or in ISPMODE. Once in the MEMLIST display, you can delete a part of the table, find data in the table, add additional members to the table, print a part of the table, store a part of the table in a data set, sort the table in different directions, and so on. For assistance with the different options available, use the HELP command, CUA functions, or enter an O command as either a primary command or as a line command.

The primary screen for MEMLIST generally shows the most commonly used member attribute information and the alternate screens show some additional data. For source members, the secondary screen adds member TTR information and an ALIASOF field that is filled in for alias members as a result of entering ALIASCHK (alias check or option 55). The primary screen fields are:

CMD	Command. Enter a line command here.
NAME	Member name. The 8-character member name.
DATA/MSG	Data field. Enter up to 8 bytes of operand information here. It is used as a command feedback field — any information in this field that is preceded by an * is ignored when a line command is entered.
VER.MOD	Version and modification level. These numbers can range from 0 through 99 and they reflect the level of the member.
CREATED	Creation date. The format is YY/MM/DD.
LAST	Modification date. The format is YY/MM/DD.
MODIFIED	Modification time. The format is HH:MM.
SIZE	Size of member in lines. Member sizes up to 65,555 display.
INIT	Initial size of member in lines. Member sizes up to 65,555 display.
ID	User ID. The user ID of the last person to modify this member.

The second source member screen reformats the data and adds the following data for each member:

ALIASOF	Alias information field. For aliases, this field contains *ALIAS until a ALIASCHK (option 55) is performed. After that point, it contains the name of the corresponding main member if it exists or *ORPHAN otherwise.
TTR	TTR address. The address of the first record for this member relative to the start of the data set.

Load members have four different screens defined. The first screen contains the linkage edit date and most information required to manipulate load members. The second screen adds more detailed information; the third screen combines the first two screens and the fourth screen can be customized and contains Change Man data.

The first screen displays:

CMD	Command. Enter a line command here.
NAME	Member name. The 8-character member name.
DATA/MSG	Data field. Enter up to 8 bytes of operand information here. It is used as a command feedback field — any information in this field that is preceded by an * is ignored when a line command is entered.
ALIASOF	Alias information field. For aliases, this field contains *ALIAS until a ALIASCHK (option 55) is performed. After that point, it contains the name of the corresponding main member if it exists or one of the following: ■ *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 noted 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
LEN/LKED	Length/LKED date. If LKEDDATE is in effect, the IDR data for each member is read and this field contains the linkage edit date in YY/MM/DD format. Otherwise, the module length displays using 6 hexadecimal characters.
UTES	Linkage attributes. This is four columns that contain linkage attributes as follows: ■ DC - downward compatible attribute ■ LOAD - load only attribute ■ NOED - NOEDIT attribute ■ NOEX - NOT Executable attribute ■ OVLY - overlay attribute ■ REFR - refreshable attribute ■ RENT - reentrant attribute ■ REUS - reusable attribute ■ SCTR - scatter attribute ■ ERRL - load only attribute conflict with scatter attribute ■ ERRO - overlay attribute conflict with reentrant, reusable, load only, or scatter attribute ■ ERRS - scatter attribute conflict with reentrant, overlay, or load only ■ ERR1 - refreshable attribute conflict with not reusable attribute
APF	This field contains AC=1 if the module is marked as authorized. If the field is incorrectly formatted it contains ERR; otherwise, this field is blank.

MODE	This field contains an indication of the residence mode of the module. If AMODE and RMODE have not been set, this field is blank; otherwise, this field contains one of the following:
	■ RANY - RMODE of ANY and AMODE 31
	■ AANY - RMODE of 24 and AMODE ANY
	■ A31 - RMODE of 24 and AMODE 31
	■ A24 - RMODE of 24 and AMODE 24
	■ A64 - RMODE of 24 and AMODE 64.
	■ ERR - RMODE of ANY; AMODE ANY or AMODE of 24. This is invalid
MAIN	This field is for alias entries. It contains the name of the corresponding main member as noted in the directory entry. Check this information for correctness by ALIASCHK (Option 55).

The second load module screen adds the following fields:

TTR	TTR address. The address of the first record for this member relative to the start of the data set.
MATCH	Matching member by TTR. This field is for alias entries. It is filled in by ALIASCHK (option 55) to show the name of the matching main member according to the TTR address. If the ALIASOF field has an unusual value (such as =BYTTR or =BYNAME), this field and the MAIN field provide names of the affected members.
LENGTH	Module length. Hexadecimal length of the module.
LEN-KB	Module length in 1024 or K-byte units.
ENTRY	Entry point address in hexadecimal.
SSI	SSI data in hexadecimal.

Save MEMLIST tables permanently. They are saved in the ISPF profile data set by default (ISPPROF) but you can change the DDNAME in the SETALL option. To manage these tables, use the SAVE and MERGE commands.

Saved MEMLIST tables are given members names of the form PDSML0na if the name is one or two numeric digits; otherwise, a name of the form @#name is used where name is a 1- to 6-character alphanumeric table identification name.

The following primary commands are supported directly for the MEMLIST function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

*	Merges current member group members into the MEMLIST table.
	```*cm ERR- t="Command,*"> ```
ALIASCHK	Alias check. Checks aliases and adds all associated members to the MEMLIST display.
ALL	MEMLIST all. Adds all members in the data set to the MEMLIST display.
ALT[ERNAT]	Displays an alternate view of the MEMLIST table.
APP[LY]	Applies the specified line command to all table entries and executes each entry.

	Syntax	`APPLY linecmd`
BAT[CHJCL]	Builds batch JCL that includes all StarTool FDM statements in the LOG table.	
DUA[L]	Displays a double line view of the MEMLIST table.	
EDITT[BL] (or ET[BL])	Enters an edit session on MEMLIST table data.	
EQ[UATE]	SUBLIST =. Resets the current member group to the members in the current MEMLIST.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	
	Syntax	`F anystring [ASIS]` `    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]` `    [ANY|MEMBER|DATE|LENGTH|ENTRY|MAIN|` `      |MATCH|SSI|TTR|ALIASOF]     `(for load)` ` [ANY|MEMBER|VERMOD|CREATED|MODIFIED|` ` |SIZE|INIT|ID|TTR|ALIASOF] `(for source)`
LE[FT]	Rotates through alternate views of the MEMLIST table. PF keys 10 and 22 are normally set to LEFT.	
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax	`LOCATE pmember|ttr|puserid|pdate|size|vermod` `            |plinkdate|hexlength|apf|attrib|pssi`
MER[GE]	Adds members to the MEMLIST table from a saved MEMLIST table.	
	Syntax	`MERGE name [RESET] [NOSTATS]`
	(where name is 1 to 6 alphanumeric characters)	
MON[TH]	Merges members updated or created this month into the MEMLIST.	
NOR[MAL]	Displays the default view of the MEMLIST table.	
O[PTIONS]	Provides primary command selection for the MEMLIST function and operand syntax assistance.	
OUT[PUT]	Outputs the MEMLIST table to print or a data set.	
	Syntax	`OUTPUT [=c | F(ddname)]`
REM[OVE]	Trims the MEMLIST table based on a string match.	

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [MEMBER|VERMOD|CREATED|MODIFIED|SIZE|INIT|`
	`        ID|TTR|ALIASOF|DATA|MSGS] (source members)`
	`        [MEMBER|DATE|LENGTH|ENTRY|MAIN|MATCH|SSI|`
	`        TTR|ALIASOF|DATA|MSGS] (load members)`	

RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the MEMLIST table. PF keys 11 and 23 are normally set to RIGHT.	
SAVE	Creates a permanent table for use in a different StarTool FDM session.	
	Syntax	`SAVE [name] [REPLACE|NOREPL]`

	(where name is 1 to 6 alphanumeric characters)	
S[ELECT]	Normally BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.	
	Syntax	`SELECT member`
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	

	Syntax
	`SORT`
	`[MEMBER|TTR|DATE|SIZE|APF|MODE|ATTR|TTR]` (load)
	`[MEMBER|TTR|VERMOD|CREATED|MODIF|SIZE|ID]`(source)
	`[ASCEND|DESCEND]`	

TAG	Forms a member group containing only members marked with *TAG* in the DATA/MSG field.	
TOD[AY]	Merges members updated or created today into the MEMLIST.	
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. The MEMLIST UT panel also allows dynamic primary commands with which you specify command names and their corresponding actions.	
WE[EK]	Merges members updated or created this week into the MEMLIST.	
X	Clears the MEMLIST table relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the MEMLIST table. This is equivalent to X ALL	

In addition to the above, you can enter StarTool FDM subcommands and ISPF commands on the command line. Also, if you want a StarTool FDM subcommand to apply to all members currently in the MEMLIST, use **=** as the member group name for the subcommand.

By default, line commands that produce output display it in an edit session and this output is also recorded in the log. To place this output only in the log, check the setting of "Automatic EDITLOG" in SETALL for MEMLIST. The following line commands are supported in the MEMLIST function:

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
ATTR	Provides a preinitialized entry assist panel to update member attributes.
CSEC	Invokes the CSECTS function for a load member.
DEC	Decrypts the member into another data set.
ENC	Encrypts the member into another data set.
INFO	Displays attributes of the member using the subcommand.
K	Kills and clears all following line commands.
LOG	Copies the line into the log.

M	Provides line command selection and entry assistance.
ML	Updates statistics and attributes for this member.
O	Provides line command selection and operand syntax assistance.
PAN	Displays this member as a panel using ISPF services.
S	Selects line command (normally BROWSE for load and EDIT for source; see SETSEL).
SWAP	Renames swap member with the member named in the DATA/MSG field.
TAG	Marks this member line with *TAG* in the DATA/MSG field.
UT	Selects the extended user line command panel.
W	Adds the data set name and the member name to the WORKPAD.
X	Drops this member from the MEMLIST table.
XMIT	Transmits this member to another user.

In addition to the above line commands, most StarTool FDM subcommands that define a memgroup can be used as line commands. This set of subcommands includes ABE, ACFCOMP, BROWSE, COPY, DCF, DELETE, DELINK, DIRENTRY, DISASM, DUP, EDIT, FIND, FSE, HISTORY, LIST, LLA, MAP, OUTCOPY, PGMDOC, PRINT, READOBJ, REPLACE, REPRO, REVIEW, SEPARATE, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT, and XREF.

Also, you can use the ALIAS, COMPARE, REPRO, and RENAME subcommands as line commands but they each require another member name to be entered as an operand in the DATA/MSG field.

Each subcommand name must be abbreviated to a maximum of four characters to fit into the CMD field. Also, several of the above subcommands are available only on source data sets (ABE, ACFCOMP, DCF, EDIT, FSE, READOBJ, SUBMIT, and TSOEDIT) and others are available only on load data sets (DELINK, DISASM, HISTORY, LLA, MAP, PGMDOC, and XREF).

MEMLIST and Deleted Members

If a MEMLIST operand is added to a RESTORE subcommand, RESTORE adds deleted members to the MEMLIST table instead of resurrecting them. For more details, see . In addition, if you set "Remove deleted members" in SETALL for MEMLIST to NO, members deleted by the DEL line command are also maintained in the MEMLIST table. Use a REST line command (with an optional new name in the DATA/MSG column) to resurrect a deleted member.

A restricted subset of line commands are available for examining and manipulating deleted members. The following line commands are supported for this purpose:

=	Equal. Repeats the previous line command. `*cm ERR- t="Line command,=">`
DELI	Delink. Delinks the deleted member (for load only).
DISA	Disasm. Disassembles the deleted member (for load only).
DUP	Duplicate. Copies the deleted member to another data set.

E	Edit. Enters edit on the deleted member. The SAVE command is not permitted.
F	Find. Searches the deleted member.
HIST	History. Displays historical data from the deleted member (for load only).
K	Kill. Cancels and clears all following line commands.
L	List. Lists the deleted member.
LOG	Log. Copies the line into the log.
M	Menu. Provides line command selection and entry assistance.
MAP	Map. Displays the structure of the deleted member (for load only).
LLA	Informs LLA that the member has been deleted.
PB	PBROWSE. Browses the deleted member.
PE	PEDIT. Enters edit on the deleted member. The SAVE command is not permitted.
REST	Restore. Resurrects this deleted member.
REV	Review. Browses the deleted member.
X	X. Drops this deleted member from the MEMLIST table.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the MEMLIST function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR- t="Block line command,==">`
AA	subcommand. Displays attributes.
BB	BROWSE subcommand. Enters ISPF browse.
DD	DIRENTRY subcommand. Displays the directory entry.
DDEL	DEL command. Deletes each member.
EE	EDIT subcommand. Enters ISPF edit (source only).
HH	HISTORY subcommand. Displays module history data (load only).
LL	LIST command. Lists each member.
MM	MAP subcommand. Displays module map (load only).
OO	O command. Provides line command assistance for each line.
PP	PRINT subcommand. Produces a hard-copy.
SS	S command. Selects a member (normally edit or browse).
TT	TAG command. Marks each member line with *TAG* in the DATA/MSG field.
VV	VERIFY subcommand. Validates each member.
XX	X command. Drops the range of table lines.

MEMLIST : RMODEANY produces the following:

```
------------------------ Load MEMLIST 1, Session# 2 -------  ROW 1 TO 5 OF 5
COMMAND ===>                                         SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=(ASID ----------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED -- UTES  -- APF  MODE  MAIN
       ASID                       92/06/15                       RANY
     LOOK                         90/12/19                  AC=1 RANY
     PDSPGM                       93/01/17 RENT REUS             RANY
     PDSPGMV        *ALIAS        93/01/17 RENT REUS             RANY PDSPGM
     WHATO                        89/02/17 RENT REUS             RANY
```

```
------------------------ Load MEMLIST 2, Session# 2 -------  ROW 1 TO 5 OF 5
COMMAND ===>                                         SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=(ASID ----------------------------
CMD  NAME     DATA/MSG ALIASOF  TTR      MAIN     MATCH     LENGTH LEN-KB  ENTRY
       ASID                     000205                      000C30    4K  000000
     LOOK                       01370F                      00D8F8   55K  000000
     PDSPGM                     00AB1E                      082D50  524K  0001A8
     PDSPGMV        PDSPGM      00AB1E PDSPGM   PDSPGM       082D50  524K  024ED0
     WHATO                      006E11                      060D88  388K  000000
```

TODAY was entered to produce the following:

```
------------------------ Source MEMLIST 1, Session# 1 ----  ROW 1 TO 16 OF 20
COMMAND ===>                                         SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(@COPY -----------------------------
CMD  NAME     DATA/MSG VER.MOD   CREATED   LAST MODIFIED  SIZE  INIT    ID
       @COPY            01.18   87/11/11  92/10/09  9:22  1813  1408  SER07
       @DSNAME          01.45   89/03/26  92/10/09 12:27  1042   694  SER07
       @DUP             01.31   92/05/01  92/10/09  8:56  1466   712  SER07
       @PDSMAIN         01.84   90/01/30  92/10/09  8:32  6143  5190  SER07
       @USAGE           01.20   87/11/11  92/10/09  7:03  1095   341  SER07
       PDSAAUX          01.05   92/07/30  92/10/09 14:56   128   122  SER07
ut     PDSMODL5         01.20   92/07/22  92/10/09  7:18   131   117  SER07
       PDSMODL6         01.12   92/07/22  92/10/09  7:18   125   117  SER07
       PDSMODL7         01.18   92/07/22  92/10/09  7:19   114    99  SER07
```

```
------------------ ut for ML (Source)   user line commands -----------------
OPTION  ===>

Choose one of the following for member PDSMODL5
    ASM  - ASMCL Clist
    ALC  - Allocate dataset and member prompt
    CDPU - Composite Document Printing Utility (GDDM application)
    CLS  - Construct user clist from prompt panel
    EQ   - Edit member with EQUAL macro
    EXEC - Invoke PDSMODL5 as a CLIST
    PREP - CLIST for Panel preprocessing using ISPPREP
    PT   - Direct command to PRINT prompt panel
    %xxx - Invoke %XXX SER07.LIB.CNTL PDSMODL5

Dynamic Commands--note:  < = ,        ! = PDSMODL5,
                         / = 'SER07.LIB.CNTL(PDSMODL5)'
NONE -  ! NONE           /* REMOVE ISPF STATISTICS
STAT -  ! ADDSTATS       /* CREATE ISPF STATISTICS
RADD -  ! NONE ADDSTATS  /* REMOVE, THEN CREATE ISPF STATISTICS
PRT_ - PRINT ! CLASS(X) FORM(DX00) ASA NOMSG
____ -
```

```
----------------------- Set PANEL Defaults ----------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
...
 MEMLIST Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES  (Yes/No) -- Load module attributes
 Show TTR, Size view   ===> YES  (Yes/No) -- TTR location and size
 Show Double line view ===> YES  (Yes/No) -- Combined information
 Show Change Man view  ===> NO   (Yes/No) -- Change Man date .. (customized?)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment      ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment      ===> 50       volumes processed between messages
 VMAP increment         ===> 500      data sets processed between messages
```

```
------------------- Load MEMLIST (Attributes), Session# 1 --- Row 1 to 5 of 5
 COMMAND ===>                                             SCROLL ===> CSR
 - DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/ ------------------------------
CMD  NAME     DATA/MSG ALIASOF LEN/LKED   -- UTES -- APF MODE  MAIN
     CMNAPSPL                  1994/06/02 RENT REUS              RANY
     DSAT                      1995/12/19 REFR RENT         AC
     DSATA          DSAT       1995/12/19 REFR RENT         AC      DSAT
     PDSE520                   1997/03/13 REFR RENT
     VTOC                      1995/06/14 REFR RENT
```

```
------------------- Load MEMLIST (TTR/Size), Session# 1 ---- Row 1 to 5 of 5
 COMMAND ===>                                             SCROLL ===> CSR
 - DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/ ----------------------------
CMD  NAME     DATA/MSG ALIASOF  TTR      MAIN     MATCH    LENGTH  LEN-KB  ENTRY
     CMNAPSPL                   027F08                     006140    25K  000000
     DSAT                       048005                     002558    10K  000000
     DSATA          DSAT        048005 DSAT      DSAT      002558    10K  000000
     PDSE520                    048C08                     0AE020   697K  0A4198
     VTOC                       034506                     004790    18K  000000
```

```
---------------------- Load MEMLIST (Double), Session# 1 ----- Row 1 to 5 of 5
 COMMAND ===>                                             SCROLL ===> CSR
 - DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/ ------------------------------
CMD NAME     DATA/MSG ALIASOF LEN/LKED   -- UTES -- APF MODE  MAIN
                      MATCH    LENGTH  LEN-KB   TTR      ENTRY    SSI
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
     CMNAPSPL                  1994/06/02 RENT REUS              RANY
                               006140    25K   027F08   000000   40BE1799
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
     DSAT                      1995/12/19 REFR RENT         AC
                               002558    10K   048005   000000
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
     DSATA          DSAT       1995/12/19 REFR RENT         AC      DSAT
                      DSAT     002558    10K   048005   000000
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
     PDSE520                   1997/03/13 REFR RENT
                               0AE020   697K   048C08   0A4198
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
     VTOC                      1995/06/14 REFR RENT
                               004790    18K   034506   000000   ABACADAE
```

```
------------------- Load MEMLIST (Change Man), Session# 1 --- Row 1 to 5 of 5
COMMAND ===>                                              SCROLL ===> CSR
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/  ----------------------------
CMD NAME      DATA/MSG ALIASOF  LEN/LKED    --  UTES   -  CMN date  time
    CMNAPSPL                    1994/06/02 RENT REUS           1994/06/02 18:08
    DSAT                        1995/12/19 REFR RENT
    DSATA            DSAT       1995/12/19 REFR RENT
    PDSE520                     1997/03/13 REFR RENT
    VTOC                        1995/06/14 REFR RENT
```

NUCMAP Function

The NUCMAP command displays the current system nucleus.

Example

NUCMAP

Syntax

NUCMAP

Aliases

NUC, NUCM, NUCMA, NUCMAP

Defaults

Required

Operands

Remarks

NUCMAP provides a way to look at the current nucleus by memory address. The CSECTS table is used.

The CSECTS ISPF table displays in response to a NUCMAP command. When you are in a CSECTS/NUCMAP display, you can delete a part of the table, sort the table in different directions, find data in the table, print a part of the table or store a part of the table in a

data set. For assistance with the different options available, use the HELP command, CUA functions or enter an O command as either a primary command or as a line command.

The CSECTS/NUCMAP table displays the names of all CSECTs and ENTRYs for the NUCLEUS. The screen fields for each line are:

CMD	Command. Enter a line command here.
TYPE	Type of entry. Any of the following may appear: CSECT – CSECT entry. ENTRY – ENTRY within a CSECT.
DATA/MSG	Data field. This is a command feedback field.
SECTION	CSECT name.
ENTRY	ENTRY name.
ADDR	Address of this symbol (this is for the CSECTS function only).
LENG	Hexadecimal length of this CSECT.
MD/SG	Mode or segment. An AMODE/RMODE indication is provided as follows: ■ R64 - RMODE of ANY and AMODE 64 ■ A64 - RMODE of 24 and AMODE 64 ■ RANY - RMODE of ANY and AMODE 31 ■ AANY - RMODE of 24 and AMODE ANY ■ A31 - RMODE of 24 and AMODE 31 ■ A24 - RMODE of 24 and AMODE 24
MEM ADDR	Memory address in hexadecimal.

The following primary commands are supported directly for the CSECTS/NUCMAP function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Chapter 2, "ISPF Interface Commands".

APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
EDITT[BL] (or ET[BL])	Enters an edit session on CSECT/NUCMAP table data.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	

	Syntax	`F anystring [ASIS]`
		`    [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
		`    [ANY|ADDRESS|NAME|ENTRY|SEGMENT|MODE]`

L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax	`LOCATE csectname|address`
O[PTIONS]	Provides primary command selection for the CSECTS/NUCMAP function and operand syntax assistance.	

OUT[PUT]	Outputs the CSECTS/NUCMAP table to print or a data set.	
	Syntax	OUTPUT [=c | F(ddname)]
REM[OVE]	Trims the CSECTS/NUCMAP table based on a string match.	
	Syntax	REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT] [ANY|ADDRESS|NAME|ENTRY|SEGMENT|MODE]
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	SORT [ADDRESS|NAME] [ASCEND|DESCEND]
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.	
	Syntax	TAG linecmd
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.	
X	Clears the CSECTS/NUCMAP table relative to the cursor position.	
	Syntax	X [ABOVE|BELOW|ALL]
XA[LL]	Clears the CSECTS/NUCMAP table. This is equivalent to X ALL	

The following line commands are supported in the CSECTS/NUCMAP function:

+	Provides an extension panel for command entry. *cm ERR- t="Line command,+">
=	Repeats the previous line command. *cm ERR- t="Line command,=">
K	Kills and clears all following line commands.
LOG	Copies the line into the log.
O	Provides line command selection and operand syntax assistance.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
X	Drops the table line.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

==	= command. Repeats the previous line command for the range of lines. *cm ERR- t="Block line command,==">
00	O command. Provides line command assistance for each line.
XX	X command. Drops the range of table lines.

```
 FUNCTIONS  CONTROL  LINE CMDS  DEFAULTS  FEATURES
--------------------------------------------------------------------------------
----------------------- CSECT Display for *NUCMAP*  --  ROW 1 TO 17 OF 3,526
COMMAND ===>                                               SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(PDSPNAX  --------------------------
CMD  TYPE   DATA/MSG SECTION  ENTRY     MEM ADDR -LENG- MODE
     CSECT           IOSVATTN           00FC3D00 000106 A24
     CSECT           IEAVFX00           00000000 001000 A24
     ENTRY                    IEATCBP   00000218        A24
     CSECT           IECVDDT5 00FCE000  00004C          A24
     CSECT           IECVPRNT 00FCE050  000478          A31
```

Chapter 7

Commands — O to R

OPTIONS Subcommand

The OPTIONS subcommand displays the subcommands (or options) supported by StarTool FDM. This display varies when a non-partitioned data set is being processed.

In any table function, if the O alias of OPTIONS is entered as a line command, the **O Subcommand Selection** panel displays to assist in entering subcommands. In ISPMODE or MEMLIST, the O alias of OPTIONS at the **COMMAND ==>** prompt also brings up the subcommand assist prompt panel.

```
---------------------- StarTool O Subcommand Selection --------------------------
OPTION  ===>

Choose one of the following:
                                                               More:     +
   --- CONTROL ---
   LEFT  - Scroll command or PF key
   RIGHT - Scroll command or PF key
   F     - Find data in table prompt
   OUT   - Output a copy of table to print or to a data set
   UT    - Extended user command panel
   X     - Clear a portion of the table
   XMIT  - Transmit this data set to another user
   LASTCMD  - display the last commands entered from the command line
   SETUSER  - define dynamic user commands and operands
   --- FUNCTIONS ---
   STATUS   - Display function status and selection panel
```

Example
OPTIONS

Syntax
OPTIONS

Aliases
O, OP, OPT, OPTI, OPTIO, OPTION, OPTIONS

Operands

No operands are supported for the OPTIONS subcommand.

OUTCOPY Subcommand

The OUTCOPY subcommand formats utility control statements that can be used by IEBCOPY to select members from a data set. Output is directed to a pre-allocated FILE with the DDNAME PDSOUT.

Example

```
OUTCOPY    mema:memb
```

Syntax

```
OUTCOPY    memgroup [ALIAS|NOALIAS         ]
                    [CLOSE|NOCLOSE         ]
                    [ECHO|NOECHO           ]
                    [IEBUPDTE|NOIEBUPDTE   ]
                    [REPLACE               ]
                    [TEXT('string')|NOTEXT]
```

Aliases

```
OUTC, OUTCO, OUTCOP, OUTCOPY
```

Defaults

initially: memgroup, ECHO, NOCLOSE, NOALIAS, NOTEXT, NOIEBUPDTE

thereafter: memgroup, NOCLOSE, previous(ALIAS|NOALIAS; ECHO|NOECHO; TEXT|NOTEXT; IEBUPDTE|NOIEBUPDTE)

Required

Operands

memgroup	Identifies the member names to be output on an IEBCOPY format select statement.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
ALIAS	Includes any associated member names (aliases, apparent aliases, and associated main members) in the members processed.
NOALIAS	Processes only members actually identified.
CLOSE	Closes the PDSOUT file. The member or member group specified with the CLOSE operand is not output to the PDSOUT file.
NOCLOSE	Opens or keeps the PDSOUT data set open.
ECHO	Echoes each utility control statement to the terminal as well as to the PDSOUT data set.

NOECHO	Does not echo the utility control statements to the terminal; output is only directed to the PDSOUT data set.
IEBUPDTE	Formats each output statement in IEBUPDTE format instead of the standard IEBCOPY format. The use of this format also suppresses COPY statements normally generated for data set changes.
NOIEBUPDTE	Formats each output statement in the standard IEBCOPY format.
REPLACE	Specifies that the form of the COPY statement is to be changed to support replacing members in the target data set.
TEXT('st')	Replaces the second member name on each subsequent SELECT output statement with the specified string, ST.
NOTEXT	Nullifies any TEXT operand and writes the second copy of the member name (as usual) on each subsequent SELECT output statement.

Remarks

OUTCOPY does not use ISPF services. If StarTool FDM was invoked from READY mode, StarTool FDM attempts to establish an ISPF environment for the first OUTCOPY subcommand so that subsequent subcommands can use ISPF services.

OUTCOPY usage notes:

1   Output is directed to FILE(PDSOUT).

2   PDSOUT must be allocated to a sequential data set or to a member of a partitioned data set with fixed, 80 character records prior to entering the first OUTCOPY subcommand.

3   For new data sets, the following default DCB attributes are used:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

4   An IEBCOPY COPY statement is written before the first SELECT statement for each data set as determined by the CHANGE subcommand. The COPY statement is of the form:

```
COPY OUTDD=OUTPUT,INDD=xxx
```

(where xxx is the low-level qualifier of the input data set name)

However, if the REPLACE keyword is in effect, the format is:

```
COPY  OUTDD=OUTPUT,INDD=((xxx,R))
```

5   The default IEBCOPY select statement is of the form:

```
S M=memname,memname))
```

memname is the member name. You can modify this format output statement by an editor if member rename or member replace functions are to be performed. If you specify TEXT('Any String'), each subsequent IEBCOPY statement changes to the form:

```
S M=memname  ANY STRING
```

6   OUTCOPY maintains a list of member names written after each copy statement and does not output a member name more than once.

7   You can establish a CLIST such as the following for IEBCOPY use:

```
PROC 3 FROMDSN FROMQUAL TODSN CONTROLS(PDSOUT.CNTL)
ALLOC F(SYSIN) DA(&CONTROLS.) SHR
ALLOC F(SYSPRINT) DA(*)
ALLOC F(&FROMQUAL.) DA(&FROMDSN.) SHR
ALLOC F(OUTPUT) DA(&TODSN.) OLD
CALL 'SYS1.LINKLIB(IEBCOPY)'
```

8   Assuming that `OUTCOPY` has created output in `PDSOUT.CNTL`, copy selected members from your `LIB.LOAD` data set to `'SYS1.LINKLIB2'`, by entering the following line from READY mode:

```
%COPYOUT LIB.LOAD LOAD 'SYS1.LINKLIB2'
```

Another method for capturing IEBCOPY control statements is to use the `COPY` subcommand with the `NOCALL` keyword. Obtain the generated images using the `CONTROL` subcommand with the `DSN` keyword or by the `ISPMODE OUTPUT` command.

PATTERN Subcommand

The PATTERN subcommand lists member names from the data set directory. Only those member names that contain the PATTERN name segments specified display. The default member group is not affected by the DISPLAY or PATTERN subcommands.

You can enter one or two member name segments as patterns. If you enter two segments, both segments must occur in a member name for it can be displayed. For example, with the pattern segments xx and 1x, members named M1X4XX, M1XX, and XX1X are displayed by this subcommand while members named X1X or MXX13X are not displayed.

Example
```
PATTERN   iea/sys
```

Syntax
```
PATTERN   [segment1 [segment2] ]
```

Aliases
```
P, PA, PAT, PATT, PATTE, PATTER, PATTERN
```

Defaults

previously entered segment1 and segment2 operands

Required

Operands

segment1	A name segment that must be found in a member name for the member name to be displayed.
	Enter a group name descriptor such as part1:part2, seg1/seg2, or part1*seg1 for segment1, in which case, the output consists of the member names that are displayed by the MEMBERS subcommand.
segment2	A second name segment that must also be found in a member name for the member name to be displayed.

Remarks

Neither PATTERN nor DISPLAY affect the current member group. If you enter PATTERN without operands, the previously entered operands are used. This differs from DISPLAY. A DISPLAY with no operands shows all members in the data set.

 -A is added following a displayed member name if the member name is an alias. The member name is listed in a combined hexadecimal/character format if it does not contain upper case alphanumeric characters (including @, $, and #) or if the first character is numeric. Unprintable characters (for 3270-type devices) in a member name display as periods.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
--------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ---------- ROW 273 OF 285
COMMAND ===>                                                SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=(ASID --------------------------
>----->d
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK     ALLGLOB     ASMEXT2     CHECKOUT    CICS
COMPARE     COMPARES    COMPA1      COMPA2      COMPA3      DISAMSM
DISAMXM     DISASME     DISASMX     DISASM3     MAPXREF     PDS99
PDS99T1D    TESTMSG
>----->p dia
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK
>----->p al/
@DIALOG     ALLGLOB
>----->p ob/al
ALLGLOB
>----->p dis*
DISAMSM     DISAMXM     DISASME     DISASMX     DISASM3
>----->p
DISAMSM     DISAMXM     DISASME     DISASMX     DISASM3
```

PBROWSE Function

PBROWSE (for parallel browse) is a multiple-format data set browser that supports most z/OS data access methods and database management systems. PBROWSE is called a parallel function because you can suspend a browse session to process other data sets using GO sessions. On returning control to the browse session, resume where you left off as with other StarTool FDM parallel activities.

Invoke PBROWSE directly under control of StarTool FDM with the PBROWSE command, or, using the SUPEREDIT option, invoke it transparently with the BROWSE command. Use PB (for PBROWSE) as a line command in MEMLIST.

PBROWSE supports sequential, direct, PDS, PDSE and all VSAM data types including PATHs. It also allows you to browse deleted members. In addition, you can perform VSAM positioning.

> 📝 **NOTE** PBROWSE does not support multivolume files. It also cannot be used with variable-length (VB or VS) file organizations. For these file types, use PVIEW or PEDIT instead of PBROWSE.

Example
```
PBROWSE
```

Syntax
```
PBROWSE    member
       * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)   ]
       * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)         ]

*NOTE: Lines with an asterisk are supported for VSAM data sets only
```

Aliases
```
PB, PBR, PBRO, PBROW, PBROWS, PBROWSE
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

Required

member

Operands

member	Identifies the member to be browsed.
FROMKEY(*ky***)**	For VSAM data sets only, *ky* is coded as the key of the first record to be accessed. This is a generic key and can be coded as x'*hexkey*'; access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with *TOKEY*. Specified for an alternate index or a key-sequenced data set only.

FROMADDRESS(*ad***)**	For VSAM data sets only, *ad* is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) - Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components - Cannot be specified if the data set is being accessed through a path - Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, *nm* is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is coded as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'; access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.
TOADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) - Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components - Cannot be specified if the data set is being accessed through a path - Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.

Remarks

Use a LOG command to suspend PBROWSE. Then move to other StarTool FDM processes; however, use any function name or subcommand that does not conflict with the command names supported for PBROWSE.

The following primary commands are supported directly in the PBROWSE function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" on page 37.

CO[LS]	Displays a columns line on the first line of the data area. The columns line remains at the top of the data display. It is useful in identifying columns to be used with the FIND command.	
	Syntax	`COLS [ON|OFF]`
F[IND]	Finds a string and positions the display start location.	
	Syntax	`FIND anystring [nn mm] [ASIS]` `        [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	(where *nn* and *mm* are column numbers)	
HEX	Sets HEX display mode on or off.	
	Syntax	`HEX [ON|OFF]`
L[OCATE]	Positions directly to the specified record number.	
	Syntax	`LOCATE recnumber`
PAD	Changes the padding character and specifies the number of pad characters desired.	
	Syntax	`PAD character|'character'|x'hex' [column]`
	Example: PAD a 45	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	

```
PBROWSE -  WSER07.VSAM.CUST.CLUSTER --------------- LINE 00000401 COL 001 080
COMMAND ===> f new word                                    SCROLL ===> CSR
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
Smith 0392    92A444510Yorba Linda,CA 92686            YYNB
Jones 0194    92A444511Minneapolis,MN 55416            YYNX
Green 1293    92A444512Denver, CO 80222                NNYD
Watson 1093   92A444513Middlesex,UB11 4AA England      NNNF
Aku 0394      92A444515Apu,  Easter Island             NNYF
Ricker 0993   92A444518Ward Hill, MA 01835             YNYF
Jackson 0494  92A444519Springfield, MO 65890           YYND
Rogers 0394   92A444520Oakland, NJ 07436               XXMG
Thomas 0692   92A444521Naples, FL 33942                AACC
Schmidt 0993  92A444522Bremen,  Germany                MMNN
Stone 1093    92A444523Blue ridge, PA 17214            VVGC
Apple 0294    92A444524Alpharetta, GA 30201            AABB
Cznski 1293   92A444525Springboro, OH 45066            YYMN
Johnson 0394  92A444529Clinton, MD 20797               AAEE
Engel 0494    92A444533New york, NY 10117              UUFF
```

```
PBROWSE - WSER07.VSAM.CUST.CLUSTER --------------- LINE 00000401 COL 001 080
COMMAND ===> f x'd48995'                                     SCROLL ===> CSR
    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+..
       .
 Smith 0392    92A444510Yorba Linda,CA 92686            YYNB
4E98A84FFFF4444FFCFFFFFFE99884D89886CC4FFFFF444444444444EEDC444444444444444444
024938003920000921444510869210395418310926860000000000008852000000000000000000
    --------------------------------------------------------------------------------
 Jones 0194    92A444511Minneapolis,MN 55416            YYNX
4D998A4FFFF4444FFCFFFFFFD899889998A6DD4FFFFF444444444444EEDE444444444444444444
016552001940000921444511149555176392B450554160000000000008857000000000000000000
    --------------------------------------------------------------------------------
 Green 1293    92A444512Denver, CO 80222                NNYD
4C98894FFFF4444FFCFFFFFFC89A8964CD4FFFFF4444444444444444DDEC444444444444444444
079555012930000921444512455559B0360802220000000000000000055840000000000000000000
    --------------------------------------------------------------------------------
 Watson 1093   92A444513Middlesex,UB11 4AA England      NNNF
4E8AA994FFFF444FFCFFFFFFD88898A8A6ECFF4FCC4C989898444444DDDC444444444444444444
061326501093000921444513494435257B42110411055731540000005556000000000000000000
    --------------------------------------------------------------------------------
```

PEDIT Function

PEDIT (for parallel edit) is a multiple-format line editor that supports most z/OS data access methods and database management systems. PEDIT is called a parallel function because you can suspend an edit session to process other data sets using GO sessions. On returning control to the edit session, resume where you left off as with other StarTool FDM parallel activities.

PEDIT supports sequential data sets, striped sequential data sets, direct files, VSAM keyed sequential data sets (including either a base cluster or a path over a cluster), other VSAM data set types, partitioned data sets (PDSs), and extended partitioned data sets (PDSEs).

Invoke PEDIT directly under control of StarTool FDM with the PEDIT command, or, using the SUPEREDIT option, invoke it transparently with the EDIT command.

Example

PEDIT

Syntax

```
PEDIT    [member|memgroup]
         [AUDIT]
         [BACKUP]
         [COPYBOOK]
         [LOAD|NOLOAD|BLOCK]
         [SELECT]
         [VIEW]
     *   [FROMKEY(key)|FROMADDRESS(address)|FROMNUMBER(record_number)]
     *   [TOKEY(key)|TOADDRESS(address)|TONUMBER(record_number)]
```

*NOTE: Parameters marked with an asterisk are supported for VSAM data sets only.

Aliases

PE, PED, PEDI, PEDIT

Abbreviations and Synonyms

CB for COPYBOOK, FADDR for FROMADDRESS, FKEY for FROMKEY, REUSE for LOAD, TADDR for TOADDRESS, TKEY for TOKEY

Defaults

LOAD (NOLOAD for non-KSDS VSAM data sets)

Required

None

Operands

member	Name of data set member to be edited. If no default data set name has been set, PEDIT will prompt for data set. If no member or memgroup is supplied, PEDIT will prompt for member name and data set. **Alternative:** *memgroup*
memgroup	Memlist or memgroup naming multiple data set members to be edited in sequence during this PEDIT session. If no default data set name has been set, PEDIT will prompt for data set. If no memgroup or member name is supplied, PEDIT will prompt for member name and data set. **Alternative:** *member*
LOAD	The entire member is to be loaded into memory for editing and rewritten completely at the SAVE command. With this option, record lengths can be modified (even for VSAM ESDS data sets) and line commands such as I (insert), C (copy), D (delete), M (move), and R (repeat) are available. **Restrictions:** VSAM data sets must be marked REUS or must be empty. Executable load members may not use the LOAD option. **Synonym:** REUSE **Alternatives:** NOLOAD, BLOCK
NOLOAD	Only changed records are kept in memory. Line commands that modify data records are not available.
BLOCK	Only changed records are kept in memory and the edit display consists of one line for each physical record. Line commands that modify data records are not available.

COPYBOOK	PEDIT prompts for a copybook member to overlay the record and format the data displayed. **Synonym:** CB
SELECT	StarTool FDM is to prompt for record selection parameters based on the data set type and only selected records display in the edit session. No line commands are supported except for the D (delete) line command for VSAM KSDS data sets.
AUDIT	StarTool FDM is to prompt for allocation parameters and a data set name for an audit data set before starting the edit session. During the edit session, StarTool FDM records a before and after image of any changed record using the current record formatting mode (character, hexadecimal, or copybook). If PEDIT is not able to output records to the audit data set, PEDIT disables the audit facility and notifies you. You can continue the PEDIT session or terminate the session with a CANCEL command.
BACKUP	For sequential data sets, indicates that StarTool FDM is to make a backup copy of the data set before placing you in edit or selective edit. This permits editing large data sets and D (delete) and I (insert) are supported. When you enter SAVE during the edit session, any changes that have been made to that point are merged into the original data set and the backup data set must be rebuilt. This process may take some time.This option also supports multi-volume sequential data sets in LOAD or SELECT modes; however, LOAD is usually not appropriate.
VIEW	This edit session is to operate with the SAVE command disabled. Records are examined only. Use the PVIEW command as shorthand for PEDIT with the VIEW keyword.
FROMKEY(key)	For VSAM data sets only, *key* is coded as the key of the first record to be accessed. This is a generic key and it can be coded as x'*hexkey*'; access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. Specify it only for an alternate index or a key-sequenced data set.
FROMADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed

FROMNUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it can only for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is coded as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'. Access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.
TOADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
TONUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.

Remarks

PEDIT supports records lengths to 32K with updates made directly to a VSAM data set.

Use a LOG command to suspend PEDIT and then go to other StarTool FDM processes. Enter any StarTool FDM function name to suspend PEDIT, but do not suspend it with a StarTool FDM subcommand. For example, enter LISTC to suspend PEDIT and go to the LISTC/LISTF function; however, if you enter REPLACE, an invalid command prompt is returned.

PEDIT is similar to ISPF edit. You can also perform VSAM positioning. Usually changes in a VSAM KSDS are made by typing over individual characters; these characters are not upper-case (however, the CAPS command is supported).

If records are moved or copied out of sequence, PEDIT marks the record with ==SEQ> in the line number column. In a similar fashion, duplicate records are marked with ==DUP> in the line number field.

The keys for these records (and for records that are moved but remain in key sequence) are unprotected so that you can change the key field. PEDIT does not permit you to save

a data set if records are out of order or if there are duplicate keys; however, you can CANCEL the session.

Record exclusions with the X line command or the EXCLUDE or X primary command are not yet supported.

PEDIT offers several basic advantages:

Audit facility	StarTool FDM can record images of records in character, hexadecimal, or copybook mode before and after changes.
Copybook mode	COBOL, Assembler or PL/I copybook data display mode is supported in addition to character and hexadecimal. The display can be in a record layout mode with one record per screen or in vertical format with multiple records on a single screen.
Extended Copybook	Dynamically selects a copybook based on record values. This supports data sets with base and segment copybook portions.
Selective EDIT	Edits only selected records in an edit session.
Parallel EDIT	Goes from one edit session to any other PEDIT edit session or StarTool FDM function.
Long records	PEDIT supports record lengths up to 32K.
Large data sets	With the NOLOAD and BLOCK keywords, PEDIT only maintains changed records in memory using an update-in-place strategy for saving changed records.
Physical records	With the BLOCK keyword, examines and updates RECFM=U, RECFM=F/FB, and RECFM=V/VB physical blocks directly. With variable blocked data sets, you can correct physical record construction errors.

For sequential data sets, LOAD is the default; however, for very large data sets you can use NOLOAD. Only changed records are retained in memory. Also, only changed data blocks are rewritten for a SAVE command.

PEDIT supports editing a PDS load member and NOLOAD is forced. If a load member is updated with PEDIT, it marks the user's USERID as the ZAP IDR data and it assumes that the first CSECT of the module was modified (PEDIT is not aware of CSECT boundaries). Editing a PDS load member is equivalent to zapping the member; however, it is much simpler to perform with an edit.

For a PDS source member, LOAD is the default; however, you can specify NOLOAD or BLOCK.

For a PDSE source member, LOAD is forced; an updated member is always entirely rewritten when it is saved. PEDIT does not currently support saving PDSE program objects.

For VSAM data sets other than KSDS data sets, NOLOAD is forced; line commands that modify the data set are not supported and only modified records are rewritten to the data set. For a variable RRDS, you can modify the lengths of individual records in a PEDIT session.

Use the COPYBOOK command to select a copybook member as follows:

1 Enter a CB command.

2   Enter a generic member name; for example: mydata.set(xyz*)

3   This displays all members matching the generic member name in a MEMLIST.

4   Examine the members with E (edit), B (browse), or CBM (CBMAP) line commands.

5   Enter S (select) to select the desired copybook member and pass its name to copybook prompt.

6   Press Enter to process the data set with the selected copybook member.

Use a copybook in normal copybook mode (displayed in layout mode with one record on a screen), extended copybook mode (using multiple copybook members selected by data values), or vertical copybook mode (multi-record displayed in tabular mode using headers from the copybook).

Extended copybook support provides for copybooks with base and segment portions. The appropriate copybook is selected dynamically based on record values. Save and manage these copybook selections with the PROFMAN application. Once PEDIT is active, you can use the PROFMAN application to position to the CB portions and type an S line command to activate an extended copybook table.

To use extended copybooks, select copybook support from PEDIT or PVIEW and specify TEMP, REDO, CREATE, or MOD on the extended copybook line near the bottom. To modify an existing extended copybook member, type its name in the copybook save name field.

Use the TEMP option to enter a temporary extended copybook table that will not be saved and use REDO to modify a temporary extended copybook table. Use CREATE to create a saved extended copybook table and MOD to modify an existing table.

If you have multiple 01 levels in different copybooks, you normally define each one as a BASE. If a file definition is split into two or more copybooks, define the first part as a BASE from one copybook member and multiple following parts from different copybooks as SEGMENT sections. There must always be at least one BASE segment. Each one can be associated with multiple SEGMENT sections.

Once you are in the extended copybook dialog, type an S line command to define the first base segment and fill in the desired elements on the copybook selection panel. Be sure to enter the type (BAS for BASE or SEG for SEGMENT) for each item.

The following line commands are supported in extended copybook mode:

S	Selects an item to be updated.
D	Deletes an item.
R	Repeats an item.
I	Inserts a null item.

Type an A primary command when you want to activate the extended copybook in PEDIT. This restarts the edit session.

The following primary commands are supported directly for the PEDIT function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" on page 37.

BOU[NDS]	Specifies default find column limits. BOUNDS OFF and BOUNDS with no operand turn-off-bounds processing. BOUNDS LIST shows current boundaries.

	Syntax	`BOUNDS nn mm| OFF	LIST`
CAN[CEL]	Terminates the edit session with no data updates.	
CB	Prompts for a COBOL, Assembler or PL/I copybook name to overlay define edit records.	

	Syntax
	`CB [ON]|[OFF]|[NEW]|[FIRST]|[NEXT]|[PREV]`
	    `[ON]      ` -- default, switches to copybook mode
	    `[OFF]     ` -- reverts to the previous PEDIT display
	    `[NEW]     ` -- prompts for a new copybook member
	    `[FIRST] ` -- switches to the first copybook with matching values
	    `[NEXT]  ` -- switches to the next base copybook that matches
	    `[PREV]  ` -- switches to the previous base copybook that matches	

CBMAP	Prompts for a COBOL, Assembler or PL/I copybook name to display independently of edit records. CBMAP also sets the default copybook data set and member for the CB line command.	
C[HANGE]	Updates a character string or a hexadecimal string.	
	Syntax	`CHANGE anystring newstring [nn mm] [ASIS] [ALL]` `                  [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	(where *nn* and *mm* are column numbers)	
	The two strings must be the same length.	
CLOSE	Same as END, updates changed records and terminates PEDIT.	
COPYBOOK	Prompts for a COBOL, Assembler or PL/I copybook name to overlay define edit records.	

	Syntax
	`COPYBOOK [ON]|[OFF]|[NEW [member]]|[FIRST]|[NEXT]|[PREV]`
	    `[ON]            ` -- default, switches to copybook mode
	    `[OFF]           ` -- reverts to the previous PEDIT display
	    `[NEW member] ` -- Sets default copybook to named member.
	    `                      ` If no member supplied, PEDIT prompts for it.
	    `[FIRST]  ` -- switches to the first copybook with matching values
	    `[NEXT]   ` -- switches to the next base copybook that matches
	    `[PREV]   ` -- switches to the previous base copybook that matches	

CREATE	Creates a new partitioned data set member from marked copy or moves data lines in an edit session.	
	Syntax	`CREATE  [member]`
	(if a member name is not specified, use extended CREATE)	
CUT	Places marked edit sessions lines into the StarTool FDM clipboard. In a later PEDIT or ISPF edit session, use PASTE to place these lines back into a session.	

	Syntax
	`CUT [mm] [ADD|REPL] [DEFAULTS]`
	    `mm       ` -- two character table name
	    `ADD        ` -- add to the current clipboard
	    `REPL       ` -- replace clipboard contents
	    `DEFAULTS ` -- display a panel for defaults	

DIS[PLAY]	Displays operands and defaults for COPYBOOK, SCREEN, VERTICAL, SHOW, and HIDE.	
END	Same as CLOSE, updates changed records and terminates PEDIT.	

F[IND]	Finds a string and positions the display start location. Search strings up to 64 characters in length are supported.

	Syntax
	`FIND anystring [nn mm] [ASIS] [ALL]`
	`        [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD]`
	`        [CHECK]` -- for copybook, searches for invalid packed decimal
	`        [NAME]` -- for copybook, positions to a copybook variable name
	`        [SAME]` -- for copybook, searches only the current record

	(where *nn* and *mm* are column numbers)
HEX	Sets HEX display mode on or off.
	Syntax `HEX [ON|OFF]`
HIDE	For copybook display or VERTICAL display, excludes named or numbered elements from the current display.
	Syntax `HIDE {ALL|varnam|gennam|n1 [TO] n2} GROUP` `      ALL` -- all variables are to be hidden `      varnam` -- exact variable name from the copybook `      gennam` -- partial variable name like FORM-DETAIL* `      n1` -- entry number for a variable from SCREEN NUMBER `      TO` -- used with n2 to indicate a range `      n2` -- a second entry number to specify a range end with TO `      GROUP` -- hides all variables from a given copybook level
KEY	Positions directly to the record with the corresponding record key.
	Syntax `KEY 'record key'	x'hexkey'`
L[OCATE]	Positions directly to a record number or duplicate record (by key).
	Syntax `LOCATE recnumber [DUP] [FIRST]`
NEXT	Positions to a following logical record. Use SET to change the default number of records to move forward if no operands are entered. NEXT 1 SET is the default. SET changes the default and causes a new record to display.
	Syntax `NEXT [num SET]|[MAX]`
OUT[PUT]	Outputs the current PEDIT edit record to print or a data set using the current display mode (character, hexadecimal, or copybook).
	Syntax `OUTPUT [=c	F(ddname)]`
PAD	Changes the padding character and specifies the number of pad characters desired.
	Syntax `PAD character|'character'|x'hex' [column]`
	Example: PAD a 45
PASTE	Places lines from the StarTool FDM clipboard into the active edit session. Use the PEDIT or ISPF edit CUT macro to place lines into the clipboard.
	Syntax `PASTE [mm] [ZERO|KEEP] [DEFAULTS]` `        mm` -- two character table name `        ZERO` -- insert clipboard contents and clear clipboard `        KEEP` -- insert clipboard contents and keep clipboard `        DEFAULTS` -- display a panel to set defaults

PREV[IOUS]	Positions to a previous logical record. Use SET to change the default number of records to move backward if no operands are entered. PREV 1 SET is the default. SET changes the default and causes a new record to display.	
	Syntax	`PREVIOUS [num SET]|[MAX]`
RCH[ANGE]	Changes a string (repeat change) and positions the display start location. PF keys 6 and 18 are normally set to RCHANGE.	
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
REPL[ACE]	Replaces a data set or member from marked copy or moves data records in an edit session.	
	Syntax	`REPLACE  [member]`
	(if a member name is not specified, use extended REPLACE)	
RESET	Clears all pending line commands without letting them execute.	
SHOW	For copybook display or VERTICAL display, includes named or numbered elements in the current display.	
	Syntax ``` SHOW {ALL|varnam|gennam|n1 [TO] n2} GROUP ONLY ALL -- all variables are to be shown varnam -- exact variable name from the copybook gennam -- partial variable name like FORM-DETAIL* n1 -- entry number for a variable from SCREEN NUMBER TO -- used with n2 to indicate a range n2 -- entry number to specify range end with TO GROUP -- selects all variables in a given copybook level ONLY -- hides all variables except those identified ```	
SORT	For a VSAM KSDS data set, sorts records into ascending order by key.	
	Syntax	`SORT`
SCREEN	Changes the copybook display to show different types of data in the data type field and the displayed data field as shown in the table below.	

	Syntax
	 ``` SCREEN [TYPE|PICTURE|OFFSET|COLUMN|NUMBER][DATA|NODATA|HEX][LEA D|NOLEAD][LEV88|NOLEV88] ```
	 This table contains the copybook screen part, the header, and an example:
	 ``` COBOL level and name LEVEL AND NAME 02 SAMPLE-FIRST data type=TYPE LENGTH TYPE 1 PAC data type=PICTURE PICTURE S99V9 data type=OFFSET OFFSET 62 data type=COLUMN COLUMN 63 data type=NUMBER NUMBER 4 displayed data=DATA DATA 2.1 displayed data=NODATA OFFSET 62 displayed data=HEX DATA F24BF1 ```	

SUB[MIT]	Submits JCL member(s) to JES for background execution. **Syntax:** SUBMIT *memgroup* [*TSO_SUBMIT_operands*]	

VERT[ICAL]	Prompts for a COBOL, Assembler or PL/I copybook name to display records in a tabular copybook format.
	Select or exclude copybook elements to display with the SHOW and HIDE commands.

	Syntax
	 ```VERTICAL [ON|OFF|NEW [member]] [DEFAULT] [NAME|NONAME]```
	 ```[PICT|NOPICT] [COL|NOCOL] [NUM|NONUM] [ONLY]```
	 ```[LEV1|NOLEV1] [LEV88|NOLEV88]```
	 ` ON -- turns on VERTICAL with the current copybook`
	 ` OFF -- turns off VERTICAL mode`
	 ` [NEW member] -- Sets default copybook to named member.`
	 `            If no member name supplied, PEDIT prompts for it.`
	 ` DEFAULT -- sets header to NAME, NOPICT, TYPE, COL, NUM`
	 ` NAME|NONAME -- controls display of variable name`
	 ` PICT|NOPICT -- controls display of variable picture`
	 ` TYPE|NOTYPE -- controls display of variable type`
	 ` COL|NOCOL -- controls display of variable start column`
	 ` NUM|NONUM -- controls display of variable reference number`
	 ` ONLY -- display only selected headers; otherwise, additive`
	 ` LEV1|NOLEV1 -- controls if the LEVEL 1 variable is displayed`
	 ` LEV88|NOLEV88 -- controls if LEVEL 88 items are displayed`

The following line commands are supported in the PEDIT function:

=	Repeats the previous line command. ` *cm ERR- t="Line command,=">`
?	Provides line command selection and operand syntax assistance.
(	Shifts a line logically to the left.
((	Shifts a block of lines logically to the left.
)	Shifts a line logically to the right.
))	Shifts a block of lines logically to the right.
A	After this entry.
B	Before this entry.
Cn	Copies line with an optional count.
COLS	Displays a formatted line for identifying display columns.
In	Inserts line with an optional count.
Mn	Moves line with an optional count.
Rn	Moves line with an optional count.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the PEDIT function:

==	= command. Repeats the previous line command for the range of lines. ` *cm ERR- t="Block line command,==">`
CC	C command. Copy block.
DD	D command. Delete block.
MM	M command. Move block.
RR	R command. Repro block.

```
PEDIT ---   WSER07.VSAM.CUST.CLUSTER --------------- LINE 0000403 COL 001 070
COMMAND ===>                                             SCROLL ===> CSR
r2 403  Green 1293     92A444512 Denver, CO 80222              NNYD
=>COLS ....+....1....+ ....2.... +....3....+....4....+....5....+....6....+....
000404  Watson 1093    92A444513 Middlesex,UB11 4AA England    NNNF
000405  Aku 0394       92A444515 Apu,  Easter Island           NNYF
000408  Rogers 0394    92A444520 Oakland, NJ 07436             XXMG
==DUP>  Rogers 0394    92A444520 Oakland, NJ 07436             XXMG
==DUP>  Rogers 0394    92A444520 Oakland, NJ 07436             XXMG
000411  Thomas 0692    92A444521 Naples, FL 33942              AACC
000414  Apple 0294     92A444524 Alpharetta, GA 30201          AABB
000415  Smith 1292     92A444528 Phoenix, AZ 85071             ABCE
==SEQ>  Cznski 1293    92A444525 Springboro, OH 45066          YYMN
```

```
PEDIT ---   WSER07.VSAM.CUST.CLUSTER --------------- LINE 0000405 COL 001 070
COMMAND ===> c x'c197a4' x'c1d7c4'                       SCROLL ===> CSR
==DUP>  Green 1293     92A444512 Denver, CO 80222              NNYD
        4C98894FFFF4444 FFCFFFFFF C89A8964CD4FFFFF444444444444444DDEC444444444
        079555012930000 921444512 455559B036080222000000000000000005584000000000
-------------------------------------------------------------------------------
=>COLS ....+....1....+ ....2.... +....3....+....4....+....5....+....6....+....
        ....+....1....+ ....2.... +....3....+....4....+....5....+....6....+....
        ....+....1....+ ....2.... +....3....+....4....+....5....+....6....+....
-------------------------------------------------------------------------------
d 0406  Watson 1093    92A444513 Middlesex,UB11 4AA England    NNNF
        4E8AA994FFFF444 FFCFFFFFF D88898A8A6ECFF4FCC4C989898444444DDDC444444444
        061326501093000 921444513 494435257B42110411055731540000000555600000000
-------------------------------------------------------------------------------
```

```
PEDIT ---   WSER07.PDSDEMO.VSAM.CLUSTER ------------- COLUMNS 000001 000035
 COMMAND ===>                                            SCROLL ===> CSR
 000001 90A44452
 LEVEL AND NAME                 LENGTH TYP DATA ....1....+....2....+....3...
01 TESTBOOK
 02 MYFILE-SKIP                    1 CHR
 02 MYFILE-IDENTIFIER             14 CHR Apple 0294
 02 MYFILE-KEY
  05 MYFILE-KEY-FIRST2             2 CHR 90
  05 MYFILE-KEY-REST               7 CHR A444524
 02 MYFILE-PLACE-OF-BUSINESS      27 CHR Alpharetta, GA 30201
 02 MYFILE-MISSION-KEYS
  05 MYFILE-KEY-ONE                1 CHR A
  05 MYFILE-KEY-TWO                1 CHR A
  05 MYFILE-KEY-THREE-FOUR         1 CHR B
  05 MYFILE-KEY-THREE-FOUR         1 CHR B
 02 FILLER                         2 CHR
  05 MYFILE-HIT-DISTANCE-ONE       3 PAC 2.1
  05 MYFILE-HIT-DISTANCE-TWO       5 PAC .02
```

```
PEDIT        WSER07.VSAM.DEMOLONG.CLUSTER ----------- COLUMNS 000001 000070
 COMMAND ===> cb                                         SCROLL ===> CSR
 ****** ******************** TOP OF DATA ********************************
 000001 A AAAAAAAA THE  REST1
 000002 B BBBBBBBB THE  REST2
 000003 C CCCCCCCC THE  REST3
 000004 D 19460228 CHEC K DATE
 ****** ******************** BOTTOM OF DATA ****************************
```

```
---------------------------------- Copybook -------------------------
COMMAND ===>

 Primary command: MAP to view the copybook member before exiting this panel.
                  IMPORT or EXPORT to retrieve or save a copybook object.

  Copybook type       ===> COPYBOOK   (COPYBOOK, EXTENDED, or VERTICAL)
Copybook or Vertical Copybook data set name:
  Data set            ===> 'WSER077.LIB.CNTL(CB002T)'
  Member name         ===> CB002T     (name of member if not included in dsn)
  Data set type       ===> MVS        (MVS, Panvalet or Librarian)

Optional:
  Language type       ===> EITHER     (COBOL, PLI, EITHER, or ASM)
  Starting name       ===>
  Starting column     ===> 1          (1 to 32760 if first variable is offset)

 Extended copybook options:
  Extended copybook   ===> NONE     (Temp, Redo, Create, Mod, Use, Import)
  Copybook save name  ===>          (3 to 6 character name used with Mod or Use
  Source library      ===> PROFILE (Profile, Group, Sample, or Other)
  Other DSNAME        ===>
```

```
----------------------- Source MEMLIST 1, Session# 2  ----- Row 1 to 17 of 17
COMMAND ===>                                              SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=CB00* -----------------------------
CMD  NAME     DATA/MSG VER.MOD CREATED     LAST  MODIFIED     SIZE  INIT  ID
     CB000X            01.00  1997/10/06  1997/10/06 11:12   178    178 WSER07
e    CB001             01.03  1997/03/18  1998/03/27  9:53    38     37 WSER07
     CB002             01.00  1997/03/18  1997/03/18 16:26     6      6 WSER07
     CB002A            01.04  1998/06/01  1998/06/01 16:55     9      9 WSER07
     CB003             01.00  1997/03/18  1997/03/18 16:27     6      6 WSER07
     CB004             01.03  1997/07/25  1997/07/25  8:37    84     84 CHECK
```

```
EDIT      WSER07.LIB.CNTL(CB001) - 01.03              Columns 00001 00072
 Command ===>                                          Scroll ===> CSR
 000007 123456  01  TESTBOOK.
 000008 123456     02 MYFILE-SKIP               PICTURE X(1).
 000009 123456     02 FILLERX                   PIC 9(11) BINARY.
 000010 123456     02 MYFILE-IDENTIFIER         PICTURE X(14).
 000011 123456     02 MYFILE-KEY.
 000012 123456       05 MYFILE-KEY-FIRST2       PIC X(2).
 000013 123456       05 MYFILE-KEY-REST         PIC X(7).
 000014 123456     02 MYFILE-PLACE-OF-BUSINESS  PIC X(27).
 000015 123456     02 TOFILE-COUNT-BINARY       PIC S9(4).
 000016 123456     02 FILLER                    PIC X.
 000017 123456     02 MYFILE-MISSION-KEYS.
 000018 123456       05 MYFILE-KEY-ONE          PIC X(1).
 000019 123456       05 MYFILE-KEY-TWO          PIC X(1).
 000020 123456       05 MYFILE-KEY-XXX          PIC S9999 COMP.
 000021 123456       05 MYFILE-KEY-THREE-FOUR   PIC X(1) OCCURS 2 TIMES.
```

```
---------------------- Source MEMLIST 1, Session# 2  ----- Row 1 to 17 of 17
COMMAND ===>                                            SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=CB00* -----------------------------
CMD  NAME     DATA/MSG VER.MOD CREATED     LAST  MODIFIED    SIZE  INIT  ID
     CB000X             01.00  1997/10/06  1997/10/06 11:12   178   178 WSER07
s    CB001    *EDIT*    01.03  1997/03/18  1998/03/27  9:53    38    37 WSER07
     CB002             01.00  1997/03/18  1997/03/18 16:26     6     6 WSER07
```

```
---------------------------------- Copybook ----------------------------------
 COMMAND ===>
  Primary command: MAP to view the copybook member before exiting this panel.

  Copybook data set name:
    Data set          ===> 'WSER07.LIB.CNTL(CB001)'
    Member name       ===> CB00*     (name of member if not included in dsn)
    Data set type     ===> MVS       (MVS, Panvalet or Librarian)
```

```
PEDIT       WSER07.VSAM.DEMOLONG.CLUSTER --------------------------------------
 COMMAND ===>                                            SCROLL ===> CSR
 RECORD       1     KEY= AAAAAAAA THE
 LEVEL AND NAME                   LENGTH TYP DATA ....1....+....2....+....3..
01 TESTBOOK
 02 MYFILE-SKIP                        1 CHR A
 02 FILLERX                            8 BIN X'C1C1C1C1C1C1C1C1'
 02 MYFILE-IDENTIFIER                 14 CHR  THE REST1
 02 MYFILE-KEY
  05 MYFILE-KEY-FIRST2                 2 CHR
  05 MYFILE-KEY-REST                   7 CHR
 02 MYFILE-PLACE-OF-BUSINESS          27 CHR
 02 TOFILE-COUNT-BINARY                4 DIS X'40404040'
 02 FILLER                            1 CHR
 02 MYFILE-MISSION-KEYS
  05 MYFILE-KEY-ONE                    1 CHR
  05 MYFILE-KEY-TWO                    1 CHR
  05 MYFILE-KEY-XXX                    2 BIN 16448
```

```
--------------------------------- Copybook --------------------------
 COMMAND ===>

  Primary command: MAP to view the copybook member before exiting this panel.
                   IMPORT or EXPORT to retrieve or save a copybook object.

  Copybook type       ===> COPYBOOK   (COPYBOOK, EXTENDED, or VERTICAL)
 Copybook or Vertical Copybook data set name:
  Data set          ===> 'WSER07.LIB.CNTL(CB001)'
  Member name       ===> CB001     (name of member if not included in dsn)
  Data set type     ===> MVS       (MVS, Panvalet or Librarian)
                                                          More:    +
Optional:
  Language type       ===> EITHER     (COBOL, PLI or EITHER)
  Starting name     ===>
  Starting column   ===> 1           (1 to 32760 if first variable is offset)

Extended copybook options:
  Extended copybook  ===> temp    (None, Temp, Redo, Create, Mod, or Use)
  Copybook save name ===>         (3 to 6 character name used with Mod or Use)
  Source library     ===> PROFILE (Profile, Group, Sample, or Other)
  Other DSNAME       ===>
```

```
--------------------------- Extended Copybook #1 ---------------------------
COMMAND ===>

 Primary command: MAP to view the copybook member before exiting this panel.
                                                             More:     +
 Specify the copybook data set and member in normal TSO syntax:
  Data set                  ===> LIB.CNTL
  Member name               ===> cb001      (member if not included in dsname)
  Data set type             ===> MVS        (MVS, Panvalet or Librarian)
  Language type             ===> EITHER     (COBOL, PLI or EITHER)
  Starting name             ===>
  Type                      ===> BASE       (The first one must be a BASE)
  Column number             ===> 1          (1-32760 or Blank for default)
  Compare operator/Length   ===> EQ         (LT,LE,EQ,GE,GT,NE or 0-32760)
  Compare Strings           ===> c'a'                       (string,string)
    AND/OR ==> OR
  Column number             ===>            (1-32760 or Blank for no compare)
  Compare operator/Length   ===> EQ         (LT,LE,EQ,GE,GT,NE or 0-32760)
  Compare Strings           ===>                            (string,string)
```

```
--------------------------- Extended Copybook -----------------------------
 COMMAND ===> a

Primary commands: S to Save this list, A to Activate this list, END to cancel.
Line commands: S to Select one line at a time and update selection options.
                                                             More:     +


    Type Member      Level 01 name                Lang Type Selection
 _  1 BAS CB001       TESTBOOK                     EITHER    YES
 _  2 BAS CB0046      UNKNOWN                      EITHER    DEFAULT
 _  3 NA                                           EITHER    NA
```

```
PEDIT       WSER07.VSAM.DEMOLONG.CLUSTER --------------------------------------
 COMMAND ===>                                           SCROLL ===> CSR
 RECORD        1     KEY= AAAAAAAA THE
 LEVEL AND NAME                 LENGTH TYP DATA ....1....+....2....+....3....+
01 TESTBOOK                       BASE: CB001     # 1
 02 MYFILE-SKIP                     1 CHR A
 02 FILLERX                         8 BIN X'C1C1C1C1C1C1C1C1'
 02 MYFILE-IDENTIFIER              14 CHR  THE REST1
 02 MYFILE-KEY
  05 MYFILE-KEY-REST                7 CHR
 02 MYFILE-PLACE-OF-BUSINESS       27 CHR
 02 TOFILE-COUNT-BINARY             4 DIS X'40404040'
```

```
PEDIT        WSER07.PDSE531.TSRC(DEM2VSAM) ---------- COLUMNS 000001 000017
 COMMAND ===> hide 4 to 6                                    SCROLL ===> CSR
 NAME  |TESTBOOK|MYFILE-SKIP|MYFILE-IDENTIFIER|MYFILE-KEY|MYFILE-KEY-FIRST2
 TYPE  |LV1 80  |CHR 1      |CHR 14           |GRP 9     |CHR 2
 COLUMN|(1:80)  |(1:1)      |(2:15)           |(16:24)   |(16:17)
 NUMBER|1       |2          |3                |4         |5
 ****** ******************* TOP OF DATA *********************************
 000001                     Test  0311                    92
 000002                     Smith 0392                    92
 000003                     Jones 0194                    92
 000004                     Green 1293                    92
 000005                     Watson 1093                   92
 000006                     Aku 0394                      92
 000007                     Ricker 0993                   92
 000008                     Jackson 0494                  92
 000009                     Rogers 0394                   92
 000010                     Thomas 0692                   92
 000011                     Schmidt 0993                  92
 000012                     Stone 1093                    92
 000013                     Apple 0294                    92
 000014                     Cznski 1293                   92
 000015                     Smith 1292                    92
 000016                     Johnson 0394                  92
 **************************** BOTTOM OF DATA *********************************
```

```
PEDIT        WSER07.PDSE531.TSRC(DEM2VSAM) ---------- COLUMNS 000001 000043
 COMMAND ===> hide 2;show 5                                  SCROLL ===> CSR
 NAME  |TESTBOOK|MYFILE-SKIP|MYFILE-IDENTIFIER|MYFILE-PLACE-OF-BUSINESS
 TYPE  |LV1 80  |CHR 1      |CHR 14           |CHR 19
 COLUMN|(1:80)  |(1:1)      |(2:15)           |(25:43)
 NUMBER|1       |2          |3                |7
 ****** ******************* TOP OF DATA *********************************
 000001                     Test  0311       Anytown, AW, 95120
 000002                     Smith 0392       Yorba Linda,CA 9268
 000003                     Jones 0194       Minneapolis,MN 5541
 000004                     Green 1293       Denver, CO 80222
 000005                     Watson 1093      Middlesex,UB11 4AA
 000006                     Aku 0394         Apu,  Easter Island
 000007                     Ricker 0993      Ward Hill, MA 01835
 000008                     Jackson 0494     Springfield, MO 658
 000009                     Rogers 0394      Oakland, NJ 07436
 000010                     Thomas 0692      Naples, FL 33942
 000011                     Schmidt 0993     Bremen,  Germany
 000012                     Stone 1093       Blue ridge, PA 1721
 000013                     Apple 0294       Alpharetta, GA 3020
 000014                     Cznski 1293      Springboro, OH 4506
 000015                     Smith 1292       Phoenix, AZ 85071
 000016                     Johnson 0394     Clinton, MD 20797
 **************************** BOTTOM OF DATA *********************************
```

PVIEW Function

The PVIEW command is shorthand for PEDIT with the VIEW keyword. PVIEW can view a data set even if it is currently being edited. PVIEW is equivalent to PEDIT with the SAVE command disabled.

Example

```
PVIEW
```

Syntax

```
PVIEW   member  [REUSE|NOLOAD|BLOCK    ]
                [COPYBOOK|CB                  ]
                [SELECT                       ]
                [AUDIT                        ]
            * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)  ]
            * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)         ]

*NOTE: Lines with an asterisk are supported for VSAM data sets only
```

Aliases

```
PV, PVI, PVIE, PVIEW
```

Abbreviations

FKEY for FROMKEY, and FADDR for FROMADDRESS, TKEY for TOKEY, and TADDR for
TOADDRESS.

Defaults

Required

Operands

See the documentation for the "PEDIT Function".

PGMDOC Subcommand

The PGMDOC subcommand lists module descriptions from a program table. The output is
a one-line description of each module in the member group.

Example

```
PGMDOC  idc*
```

Syntax

```
PGMDOC memgroup
```

Aliases

PG, PGM, PGMD, PGMDO, PGMDOC

Defaults

memgroup

Required

Operands

memgroup	Identifies the member names of interest.
	Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names" on page 421.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ------------  ROW 1 OF 59
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SYS2.TSO.CMDLIB,VOL=SER=SYS804  MEM=(JBB2217  ------------------------
>----->pgm (jbb2217,dsnttttt,xyz456,isr*)
PDS235I  PRODUCT  FROM                 DESCRIPTION
JBB2217  MVSFMID  IBM                  MVS/SP R2.1.7            5752***** R217
DSNTTTTT MVS      IBM                  DB2 Prefix
XYZ456   UNKNOWN                       (THIS MODULE NAME WAS NOT FOUND)
ISRFR77  ISPF/PDF IBM                  3277 French Translate Table
ISRFR77A ISPF/PDF IBM                  3277 French APL Translate Table
ISRFR78  ISPF/PDF IBM                  3278 French Translate Table
ISRFR78A ISPF/PDF IBM                  3278 French Text Translate Table
ISRFR78T ISPF/PDF IBM                  3278 French APL Translate Table
ISRGE78  ISPF/PDF IBM                  3278 German Translate Table
ISRGE78A ISPF/PDF IBM                  3278 German APL Translate Table
ISRHA78H ISPF/PDF IBM                  5550 Korean Translate Table
ISRKA78K ISPF/PDF IBM                  5550 Japanese Translate Table
ISRLALIN ISPF/PDF IBM                  LMF - All Listing Interface
ISRLAUCL ISPF/PDF IBM                  LMF - Activity Log Cleanup
ISRLCFCL ISPF/PDF IBM                  LMF - Control File Control
```

PRINT Subcommand

The PRINT subcommand prints a hardcopy list of a member. Use the TSO PRINTDS or PRINTOFF command (as generated during StarTool FDM installation). Add any PRINTDS/PRINTOFF operands after the member name.

Example

```
PRINT   mema:memb  form(0012)  asa
```

Syntax

```
PRINT   memgroup [ASA             ]
                 [ASIS|CAPS       ]
                 [CLASS(c)        ]
                 [COPIES(num)     ]
                 [DEST(destname)  ]
                 [FCB(fcbname)    ]
                 [FORMS(formname) ]
                 [HEADING|NOHEADING]
                 [HOLD|NOHOLD     ]
                 [LIST|NOLIST     ]
                 [NOMSG           ]
                 [PRINT|NOPRINT   ]
                 [PROG(progname)  ]
                 [SNUM            ]
                 [TEXT            ]
                 [UCS(ucsname)    ]
                 [VOLUME(volname) ]
```

> Note: the above operands are for the PRINTOFF command.

Aliases

```
PR, PRI, PRIN, PRINT
```

Defaults

memgroup, CLASS(A), NOHOLD, LIST, PRINT, ASIS, HEADING

Required

Operands

memgroup	Identifies the members to be printed.
	Default member names, member lists, member name ranges, and member name patterns are allowed; for more information, see Appendix A, "Formatting Member Names" on page 421.
	If the memgroup is specified as a **:** (for all members), the PRINT command is called only once to print the entire data set.
others	Optional. May include any desired TSO PRINTOFF or PRINTDS operands.

Remarks

Either the TSO PRINTDS or PRINTOFF command is used. This choice was made during StarTool FDM installation. To determine which program is used, enter a CONTROL DEFAULT subcommand and look for an output line beginning "PRINT calls". This interface is optional and should only be used with the TSO PRINTDS command or the public domain PRINTOFF command.

PROFMAN Function

The PROFMAN (profile manager) command manages saved MEMLIST, LISTC/LISTF and WORKPAD tables. Primary commands such as SORT, F (find), REMOVE, and OUTPUT are useful in managing saved tables.

Syntax
PROFMAN

Aliases
PRO, PROF, PROFM, PROFMA, PROFMAN

Defaults

Required

Operands

Remarks

PROFMAN builds and displays a table containing all of your saved MEMLIST, LISTC/LISTF and WORKPAD tables. From this display, you can inspect, manage, and activate any of these tables.

The alternate (reached with a LEFT or RIGHT command) view of this function shows the first line of each saved table. This should give you some context to help you identify a given table.

The following primary commands are supported directly for the PROFMAN function. For documentation on ISPMODE commands available anywhere in StarTool FDM, see "Common Commands" on page 37.

ALT[ERNAT]	Displays an alternate view of the PROFMAN table.	
DUA[L]	Displays a double line view of the PROFMAN table.	
EDITT[BL] (or ET[BL])	Enters an edit session on PROFMAN table data.	
F	Finds a string and positions the display start location.	
	Syntax	`F anystring [ASIS]` `    [ANY|FUNC|NAME|MODIFIED|ID|SIZE|ENTRY]`
LE[FT]	Rotates through alternate views of the PROFMAN table. PF keys 10 and 22 are normally set to LEFT.	
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax	`LOC {LC|ML|WO[partname]}name|partdate|user|size`
NOR[MAL]	Displays the default view of the PROFMAN table.	
O[PTIONS]	Provides primary command selection for the PROFMAN function and operand syntax assistance.	
OUT[PUT]	Outputs the PROFMAN table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the PROFMAN table based on a string match.	
	Syntax	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]` `         [ANY|FUNC|NAME|MODIFIED|ID|SIZE|ENTRY]`
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the PROFMAN table. PF keys 11 and 23 are normally set to RIGHT.	
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [FUNC|NAME|MODIFIED|ID|SIZE]` `     [ASCEND|DESCEND]`
X	Clears the PROFMAN relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the PROFMAN table. This is equivalent to X ALL	

The following line commands are supported in the PROFMAN function:

=	Repeats the previous line command.
D	Displays this saved table.
DEL	Deletes this saved table.
K	Kills and clears all following line commands.

LOG	Copies the line into the log.
MERG	Merges this table into a MEMLIST, LISTC/LISTF, or WORKPAD table.
O	Provides line command selection and operand syntax assistance.
REN	Renames this saved table.
S	Selects line command (activates the MEMLIST, LISTC/LISTF, or WORKPAD table).
X	Drops the table line.

```
---------------------------- PROFMAN Table List ---------  ROW 1 TO 16 OF 16
COMMAND ===> right                                         SCROLL ===> CSR
Enter an ISPF command or a special control code:
   ----------------------------------------------------------------------------
CMD FUNC   NAME  DATA/MSG  Last Modified   SIZE  ID        First entry in table
     LC  ABC              93/12/08 15.00      1 WSER07    WSER07.LIB.CLIST
     LC  ISPF23           94/05/27 10.29      5 WSER07    WSER07.ISPF23.ISPLLIB
     LC  PDSE             94/08/04 08.16     14 WSER07    WSER07.LIB.PDSE
     LC  SDSF             94/08/19 09.48      9 WSER07    WSER07.SDSF.DIA410
     LC  SHARE            94/09/07 13.40     43 WSER07    WSER07.FILE020.DATA
     LC  TRYIT3           94/06/24 11.42      3 WSER07    WSER07.LIB.CLIST
     LC  WSER07           94/06/03 15.04    147 WSER07    SYS1.CS2.UCAT.VSTR80M
     LC  WSER09           94/06/03 15.05     44 WSER07    WSER09.ALL.ASM
     LC  10               94/01/10 07.23      3 WSER07    SYS1.LINKLIB
     LC  20               94/07/12 13.10      1 WSER07    WSER07.LIB.ASM
     LC  99               94/05/12 11.41     12 WSER07    WSER07.LIB.PDSE
     ML  SPEC             94/07/06 09.26     12 WSER07    ##BCBC   ##ISPF23 ##PD
     ML  TRY4             94/07/21 07.23      3 WSER07    PDSLCPNL PDSMLPNL PDSW
     WO  AAAA             94/07/06 16.27      1 WSER07    - this is a test
     WO  2                94/05/03 15.39      0 WSER07
     WO  10               94/01/04 12.33      1 WSER07    D 'SYS1.LINKLIB'
***************************** BOTTOM OF DATA ******************************
```

```
---------------------------- PROFMAN Table List ----------  ROW 1 TO 9 OF 16
COMMAND ===>                                               SCROLL ===> CSR
Enter an ISPF command or a special control code:
  -----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified   SIZE   ID
        --- First entry in table --------------------------------------------
     LC  ABC              93/12/08 15.00      1 WSER07
         WSER07.LIB.CLIST                            STR840 PO FB    80  9040
     LC  ISPF23           94/05/27 10.29      5 WSER07
         WSER07.ISPF23.ISPLLIB                       MIGRAT
     LC  PDSE             94/08/04 08.16     14 WSER07
         WSER07.LIB.PDSE                             STR84V PO FB    80  9040
     LC  SDSF             94/08/19 09.48      9 WSER07
         WSER07.SDSF.DIA410                          STR80L PS FBA  121 23474
     LC  SHARE            94/09/07 13.40     43 WSER07
         WSER07.FILE020.DATA                         STR826 PO FB    80  9040
     LC  TRYIT3           94/06/24 11.42      3 WSER07
         WSER07.LIB.CLIST                            STR911 PO FB    80  9040
     LC  WSER07           94/06/03 15.04    147 WSER07
         SYS1.CS2.UCAT.VSTR80M                       STR80M VS U      0  4096
     LC  WSER09           94/06/03 15.05     44 WSER07
         WSER09.ALL.ASM                              MIGRAT
     LC  10               94/01/10 07.23      3 WSER07
         SYS1.LINKLIB                                ES2RES PO U      0 23200
```

```
----------------------------- PROFMAN Table List --------- ROW 1 TO 16 OF 16
COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command or a special control code:
   ------------------------------------------------------------------------
CMD FUNC   NAME  DATA/MSG  Last Modified   SIZE  ID       First entry in table
     WO   AAAA             94/07/06 16.27      1 WSER07   - this is a test
     LC   ABC              93/12/08 15.00      1 WSER07   WSER07.LIB.CLIST
     LC   ISPF23           94/05/27 10.29      5 WSER07   WSER07.ISPF23.ISPLLIB
     LC   PDSE             94/08/04 08.16     14 WSER07   WSER07.LIB.PDSE
     LC   SDSF             94/08/19 09.48      9 WSER07   WSER07.SDSF.DIA410
     LC   SHARE            94/09/07 13.40     43 WSER07   WSER07.FILE020.DATA
     ML   SPEC             94/07/06 09.26     12 WSER07   ##BCBC   ##ISPF23 ##PD
     LC   TRYIT3           94/06/24 11.42      3 WSER07   WSER07.LIB.CLIST
     ML   TRY4             94/07/21 07.23      3 WSER07   PDSLCPNL PDSMLPNL PDSW
     LC   WSER07           94/06/03 15.04    147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC   WSER09           94/06/03 15.05     44 WSER07   WSER09.ALL.ASM
     WO    2               94/05/03 15.39      0 WSER07
     LC   10               94/01/10 07.23      3 WSER07   SYS1.LINKLIB
     WO   10               94/01/04 12.33      1 WSER07   D 'SYS1.LINKLIB'
     LC   20               94/07/12 13.10      1 WSER07   WSER07.LIB.ASM
     LC   99               94/05/12 11.41     12 WSER07   WSER07.LIB.PDSE
****************************** BOTTOM OF DATA ******************************
```

```
----------------------------- PROFMAN Table List --------- ROW 1 TO 19 OF 21
COMMAND ===> locate 94/06                                    SCROLL ===> CSR
Enter an ISPF command or a special control code:
   ------------------------------------------------------------------------
CMD FUNC   NAME  DATA/MSG  Last Modified   SIZE  ID       First entry in table
     WO    1               94/10/07 11.19      6 WSER07   - xxx
     WO   44               94/09/13 14.43      0 WSER07
     LC   SHARE            94/09/07 13.40     43 WSER07   WSER07.FILE020.DATA
     LC   SDSF             94/08/19 09.48      9 WSER07   WSER07.SDSF.DIA410
     LC   PDSE             94/08/04 08.16     14 WSER07   WSER07.LIB.PDSE
     ML   TRY4             94/07/21 07.23      3 WSER07   PDSLCPNL PDSMLPNL PDSW
     WO   NEW1             94/07/20 12.55      6 WSER07   - xxx
     LC   20               94/07/12 13.10      1 WSER07   WSER07.LIB.ASM
     WO   AAAA             94/07/06 16.27      1 WSER07   - this is a test
     WO   AB               94/07/06 14.42      5 WSER07   -
     WO   F                94/07/06 14.39      5 WSER07   -
     ML   SPEC             94/07/06 09.26     12 WSER07   ##BCBC   ##ISPF23 ##PD
     LC   WSER07           94/06/03 15.04    147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC   ISPF23           94/05/27 10.29      5 WSER07   WSER07.ISPF23.ISPLLIB
     LC   99               94/05/12 11.41     12 WSER07   WSER07.LIB.PDSE
     WO    2               94/05/03 15.39      0 WSER07
     LC   10               94/01/10 07.23      3 WSER07   SYS1.LINKLIB
```

```
----------------------------- PROFMAN Table List -------- ROW 15 TO 21 OF 21
COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command or a special control code:
   ------------------------------------------------------------------------
CMD FUNC   NAME  DATA/MSG  Last Modified   SIZE  ID       First entry in table
     LC   WSER07           94/06/03 15.04    147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC   ISPF23           94/05/27 10.29      5 WSER07   WSER07.ISPF23.ISPLLIB
     LC   99               94/05/12 11.41     12 WSER07   WSER07.LIB.PDSE
     WO    2               94/05/03 15.39      0 WSER07
     LC   10               94/01/10 07.23      3 WSER07   SYS1.LINKLIB
     WO   10               94/01/04 12.33      1 WSER07   D 'SYS1.LINKLIB'
     LC   ABC              93/12/08 15.00      1 WSER07   WSER07.LIB.CLIST
****************************** BOTTOM OF DATA ******************************
```

QUIT Subcommand

The QUIT subcommand terminates StarTool FDM.

Example
```
QUIT
```

Syntax
```
QUIT
```

Aliases
```
Q, QU, QUI, QUIT
```

Operands

No operands are supported for the QUIT subcommand.

Remarks

The QUIT subcommand is equivalent to multiple END subcommands. Use it from ISPMODE or MEMLIST instead of using repeated ENDs when you want to terminate StarTool FDM and not just the current function.

READOBJ Subcommand

The READOBJ subcommand reads and disassembles object code.

Example
```
READOBJ    PTF1107
```

Syntax
```
READOBJ memgroup [FLOAT|NOFLOAT          ]
                 [MVS370|MVSXA|ESA370    ]
                 [PRIV|NOPRIV            ]
                 [REASM                  ]
```

Aliases
```
READ, READO, READOB, READOBJ
```

Defaults

memgroup, NOFLOAT, NOPRIV, ESA370

Required

Operands

memgroup	Identifies the members to be disassembled.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
FLOAT	Includes floating-point instructions in the instruction set.
NOFLOAT	Does not decode floating-point instructions.
MVS370	Uses only instructions defined on MVS/370 systems.
MVSXA	Uses only instructions defined on MVS/XA systems.
ESA370	Uses only instructions defined on MVS/ESA systems. This instruction set also includes all MVS/XA instructions.
PRIV	Includes privileged instructions in the instruction set.
NOPRIV	Does not decode privileged instructions.
REASM	Outputs the disassembled instructions in a format that can be reassembled to recreate the decoded module. When you request this format, the CSECT offset field (the first output field) is placed at the end of each decoded statement.

Remarks

READOBJ uses an algorithm that allows several object decks to be present in a single member. READOBJ formats its output as follows:

CSECT offset	A hexadecimal offset in the current CSECT.
	With REASM format, this is the last output field.
name	An external name or generated internal name (format: "A" followed by a 6-character offset in the module).
operation	A 1- to 5-byte reconstructed operation code.
operands	The reconstructed operand values.
comments	For SVC calls, the original MACRO (for example, GETMAIN or TPUT).
hex representation	1 to 3 columns of the original hexadecimal data.
character representation	1 to 8 bytes of the same data in character mode surrounded by asterisks.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log# 1   ROW 370 TO 387 OF 1,370
COMMAND ===>                                            SCROLL ===> CSR
- DSN=SER07.LIB.DATA,VOL=SER=SER005  MEM=COMPAREC -------------------------
>----->readobj comparec
     00000 COMPARE  CSECT
     00000 COMPRPCL B    140(,R15)              47F0 F08C        *.00.*
     00004         BSM   R12,R3                 0BC3             *.C*
     00006         DC    C'OMPARE '             D6D4D7C1D9C54040 *OMPARE  *
     0000E         DC    C'  06/03'             404040F0F661F0F3 *  06/03*
     00016         DC    C'/92 13.4'            61F9F240F1F34BF4 */92 13.4*
     0001E         DC    C'8 '                  F840             *8 *
     00020         B     32(,R10)               47F0 A020        *.0..*
     00024         DC    X'ABABABABABABABAB'    ABABABABABABABAB *........*
     0002C         DC    X'ABABABABABABABAB'    ABABABABABABABAB *........*
```

RECALL Subcommand

The RECALL (or RC) subcommand displays the previous subcommand or displays and reenters the previous subcommand for execution.

The following subcommands are ignored for the purposes of the RECALL subcommand: CHANGE, HELP, K (KLEAR), and TSO.

Example
RECALL   enter

Syntax
RECALL   [ENTER|<u>NOENTER</u>|CHANGE]

Aliases
R, RC, RE, REC, RECA, RECAL, RECALL

Defaults

NOENTER

Required

Operands

ENTER	Displays and reenters the previous subcommand.

NOENTER	Displays the previous subcommand for typing over.
CHANGE	Generates a prototype CHANGE subcommand for the currently allocated data set.

Remarks

In ISPMODE additional options are available to retrieve subcommands from the log. LASTCMDS allows you to display or select any of the last 32 primary subcommands entered.

RENAME Subcommand

The RENAME subcommand changes a member's name.

If you are planning to rename members in a LLA controlled library, see "LLA and Deleted Members" in Chapter 6, "Commands — L to N".

Example

```
RENAME    thismem   thatmem
```

Syntax

```
RENAME    curname newname   [GROUP|NOGROUP ]
                            [SWAP|NOSWAP   ]
                            [REPBY(repnm)  ]
                            [FORCE         ]
```

Aliases

```
REN, RENA, RENAM, RENAME
```

Defaults

NOSWAP, NOGROUP

Required

curname, newname

Operands

curname	Specifies the current name of the member.
newname	Specifies the new name for the member. This new name becomes the default member name.

GROUP	Renames a group of members. The range of names affected is defined by the first operand and the resulting member name is obtained by replacing the first characters of the member name by the second group of characters from the RENAME subcommand.
	For example, if you enter RENAME IEB ABCD GROUP, all members beginning IEB... are renamed to ABCD... and a member named IEB12 is assigned the name ABCD12. Similarly, with RENAME IEB OX GROUP, a member named IEB12 is assigned the name OX12.
NOGROUP	Renames only the member specifically identified.
SWAP	Exchanges directory entries for curname and newname. Both members must exist before the operation takes place. This is the equivalent of the following three subcommands: **RENAME curname tempnam** **RENAME newname curname** **RENAME tempnam newname**
NOSWAP	Does not exchange member directory entries.
REPBY(repnm)	Replaces curname with repnm after renaming curname to newname. For example, RENAME ONE TWO REPBY(NEWONE) is equivalent to the following two subcommands: **RENAME ONE TWO** **RENAME NEWONE ONE**
FORCE	Renames even if an edit session is currently using the member.

Remarks

If you rename a main load member, any associated alias directory entries are updated to reflect the new base member name.

If you rename a main load member to itself, any associated alias entries are checked and updated as needed. A rename of a main member to itself may be useful in completing a rename sequence for a utility such as ISPF 3.1 or TSO RENAME which leave alias entries pointing at the original main member name.

> **NOTE** This subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection".

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 286 OF 299
COMMAND ===>                                             SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=@OLD:@OLD -----------------------
>----->ren @dia @old group
PDS161I Members to be renamed are: @DIACLIK, @DIAINIT, @DIALOG, @DIAPANL,
        @DIAPART, @DIASTAK, @DIATBL, @DIAWRK

PDS391A Should these members be renamed (Yes/No) ?
>----->y
PDS090I @DIACLIK has been renamed to @OLDCLIK
PDS090I @DIAINIT has been renamed to @OLDINIT
PDS090I @DIALOG has been renamed to @OLDLOG
PDS090I @DIAPANL has been renamed to @OLDPANL
PDS090I @DIAPART has been renamed to @OLDPART
PDS090I @DIASTAK has been renamed to @OLDSTAK
PDS090I @DIATBL has been renamed to @OLDTBL
PDS090I @DIAWRK has been renamed to @OLDWRK
***************************** BOTTOM OF DATA  *******************************
```

REPLACE Subcommand

The REPLACE subcommand changes portions of a member that contains a search string. Only those portions of the member that contain the search string display after a character string substitution. For load modules or VSAM data sets, the search and replace strings must have the same length.

Example

```
REPLACE  mema:memb  /before/after/  write
```

  (or equivalently)
```
REPLACE  mema:memb  'before'  'after'  write
```

  (or equivalently)
```
REPLACE  mema:memb  xc2c5c6d6d9c5xc1c6e3c5d9x  write
```

Syntax

```
REPLACE   memgroup
          +stra+strb+
          [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP  ]
          [CAPS|ASIS|IGNORE|PICTURE                ]
          [STATS|NOSTATS                           ]
          [WRITE|NOWRITE|UPDATE|NOUPDATE           ]
          [WORD|PREFIX|SUFFIX                       ]
          [OFFSET(Hx) MODULE({* | Fullm | Partm*)} ]
          [AFTER(num) DO(num) MAXMEMBERS(num) COLS(from:to) ]
          [AND('string2') ACOLS(from:to) ACAPS|AASIS|AIGNORE|APICTURE]
          [ORx('string3') OCOLS(from:to) OCAPS|OASIS|OIGNORE|OPICTURE ]
          [FORMAT(from:to,from:to, ...) ]
          [SKIPREC(n) MAXIN(n) MAXOUT(n)           ]
```

```
                    [SKIPCOL(n) MAXLEN(n) MAXFIND(n)          ]
                    [MEMBERS | MEMLIST | ML | NEWML | SUBLIST]
              *  [FROMKEY(key)|FROMADDRESS(add)|FROM (num)]
              *  [TOKEY(key)|TOADDRESS(add)|TONUMBER       ]
```

```
*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases

```
REP, REPL, REPLA, REPLAC, REPLACE
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

memgroup, previous stringa, IGNORE, NOWRITE, STATS, NUM or previous REPLACE/ FIND/LIST format

Required

Operands

memgroup	Identifies the members to be scanned and updated.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
stringa	Defines the string to be searched. It is coded as a delimited string like +searched for+. If the string is not entered or is null, the last string entered is used. The string can contain one to 32 characters. If you enter the ASIS keyword, neither string is translated to upper-case letters.
	For LBLOCK, LDUMP, BLOCK, and DUMP formats, the search and replace strings must have the same lengths.
	As an alternative, enter both strings as hexadecimal strings delimited with x's like x0123456789xabcdefx. Each string can contain 1 to 64 characters and that x333x1x and x0333x01x are equivalent.

stringb	Defines the replacement string. It is coded as a delimited string like +replaced with+. If the string is not entered or is null, the string is assumed to be null. The string can contain 0 to 32 characters. If you enter the ASIS keyword, neither string is translated to upper-case letters. For LBLOCK, LDUMP, BLOCK, and DUMP formats, the search and replace strings must have the same lengths. As an alternative, enter both strings as hexadecimal strings delimited with x's like x0123456789xabcdefx. Each string can contain 1 to 64 characters and that x333x1x and x0333x01x are equivalent.
NUM	Examines the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record. If the line number field is not numeric, switch formatting to NONUM mode for the remainder of the member. For ISPF-saved members, the high-order 6 digits of the line number field is formatted; otherwise, the low-order 6 digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from the logical record.
SNUM	Discards the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise). Searches and updates up to 256 bytes from a logical record.
NONUM	Searches or updates up to 256 bytes from a logical record (without regard to line numbers).
LBLOCK	Formats logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks. Only formatted segments that contain the search string are updated and displayed; however, a following segment also displays if the string spans a segment boundary. For load modules, only CSECT data is displayed or updated. If you enter a name in a MODULE operand, only those CSECT or ENTRY names that satisfy the condition are displayed or updated. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks. For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + is placed after the key); otherwise, it is a relative byte address.

LDUMP	Formats logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. Only formatted segments that contain the search string are updated and displayed; however, a following segment displays if the string spans a segment boundary.
	For load modules, only CSECT data is displayed or updated. If you enter a name in a MODULE operand, only those CSECT or ENTRY names that satisfy the condition are displayed or updated. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks.
	For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + is placed after the key); otherwise, it is a relative byte address.
BLOCK	Format physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks.
	Only formatted segments that contain the search string are updated and displayed; however, a following segment also displays if the string spans a segment boundary.
	For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + is placed after the key); otherwise, it is a relative byte address.
	If control interval processing is being performed, the first field is always the control interval relative byte address.
DUMP	Formats physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.
	Only formatted segments that contain the search string are updated and displayed; however, a following segment also displays if the string spans a segment boundary.
	For VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS. Up to 8 bytes of the key for a KSDS (if the key is longer than 8 characters, a + is placed after the key); otherwise, it is a relative byte address.
	If control interval processing is being performed, the first field is always the control interval relative byte address.
CAPS	Translates character string arguments to upper-case letters.
ASIS	Does not translate character string arguments to upper-case letters.
IGNORE	Searches for upper- and lower-case string arguments.

PICTURE	If specified, searches for data using matching characters and the following picture characters:	
	■ Use = for any character	
	■ Use % for any alphanumeric character	
	■ Use @ for any alphabetic character	
	■ Use # for any numeric character	
	■ Use $ for any special character	
	■ Use ~ for any non-blank character (the "not" character can also be used)	
	■ Use . for any invalid character	
	■ Use - for any non-numeric character	
	■ Use < for any lower-case alphabetic character	
	■ Use > for any upper-case alphabetic character	
	Replacement data can use characters and the following special picture characters:	
	■ Use = for any character	
	■ Use < for lower-case translation	
	■ Use > for upper-case translation	
WORD	Searches for strings preceded and followed by a non-alphanumeric character. Use this to limit the string hits.	
PREFIX	Searches for strings preceded by a non-alphanumeric character. Use this to limit the string hits.	
SUFFIX	Searches for strings followed by a non-alphanumeric character. Use this to limit the string hits.	
WRITE	Updates the member; this is an update-in-place operation.	
NOWRITE	Does not update the member; this is a trial update operation.	
UPDATE	Updates the member; this is an update-in-place operation.	
NOUPDATE	Does not update the member; this is a trial update operation.	
STATS	Updates ISPF statistics or a ZAP IDR record if any member data is changed.	
NOSTATS	Does not update ISPF statistics or a ZAP IDR record even if some data is changed.	
OFFSET(Hx)	Specifies a 1- to 6-digit hexadecimal module offset at which the search is to begin. This operand is only defined for load modules and only with the LDUMP or LBLOCK output formats. If you enter MODULE and OFFSET keywords, the offset applies to each module selected.	

MODULE(name)	Specifies a 1- to 8-byte partial external name that limits CSECT or ENTRY names to be searched or updated. This operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.
	If you enter MODULE and OFFSET keywords, the offset applies to each module selected. The MODULE operand has several valid forms:
	■ MODULE(*) – uses the previous name entered on any MODULE keyword
	■ MODULE(Fullm) – searches or updates only a CSECT or ENTRY named FULLM
	■ MODULE(Partm*) – searches or updates only a CSECT or ENTRY named PARTM...
AFTER(n)	*n* is coded as 0 through 9999999 to specify the number of matches to skip before performing replacements.
MAXMEMBERS(n)	*n* is coded as 0 through 9999999 to specify the maximum number of members to be selected by REPLACE.
DO(n)	*n* is coded as 0 through 9999999 to specify the maximum number of lines to output for a member.
COLS(from:to)	Column range to search for the primary string (COLS is only supported for formats NUM, SNUM, and NONUM). For example, COLS(1:10) specifies that the string can begin in columns 1 through 10.
AND('str2')	Another string to search in addition to the primary string. AND strings are only supported for formats NUM, SNUM, or NONUM. REPLACE is reported for a primary string only if the AND string is found first.
ACOLS(from:to)	Column range to search for the AND string. For example, ACOLS(1:10) specifies that the string can begin in columns 1 through 10.
ACAPS	If you use an AND character string argument, it is translated to upper-case letters.
AASIS	If you use an AND character string argument, it is not translated to upper-case letters.
AIGNORE	If you use an AND character string argument, the search is for upper- and lower-case data matching the string.

APICTURE	If you use an AND character string argument, the search is for data with matching characters and the following picture characters:
	■ Use = for any character
	■ Use % for any alphanumeric character
	■ Use @ for any alphabetic character
	■ Use # for any numeric character
	■ Use $ for any special character
	■ Use ~ for any non-blank character (the "not" character can also be used)
	■ Use **.** for any invalid character
	■ Use - for any non-numeric character
	■ Use < for any lower-case alphabetic character
	■ Use > for any upper-case alphabetic character
ORx('str3')	Another string to search as an alternate to the primary string. ORx can be OR2 through OR7 for a total of 8 OR conditions. OR strings are only supported for formats NUM, SNUM, or NONUM. REPLACE is reported for a primary string or an OR string.
OCOLS(from:to)	Column range to search for the OR string. For example, OCOLS(1:10) specifies that the string can begin in columns 1 through 10.
OCAPS	If you use an OR character string argument, it is translated to upper-case letters.
OASIS	If you use an OR character string argument, it is not translated to upper-case letters.
OIGNORE	If you use an OR character string argument, the search is for upper- and lower-case data matching the string.
OPICTURE	If you use an OR character string argument, the search is for data with matching characters and the following picture characters:
	■ Use = for any character
	■ Use % for any alphanumeric character
	■ Use @ for any alphabetic character
	■ Use # for any numeric character
	■ Use $ for any special character
	■ Use ~ for any non-blank character (the "not" character can also be used)
	■ Use . for any invalid character
	■ Use - for any non-numeric character
	■ Use < for any lower-case alphabetic character
	■ Use > for any upper-case alphabetic character

FORMAT (from:to,from:to, …)	Specifies the record columns to display in the output when reporting a REPLACE string. FORMAT is only supported with NUM, SNUM, or NONUM data. For example, FORMAT(21:30,0,1:10) specifies that the output is formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.
SKIPREC(n)	Ignores *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.
MAXIN(n)	Inputs up to *n* (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.
MAXOUT(n)	Displays up to *n* (coded as 0 through 9999999) output lines for a member.
MAXFIND(n)	Locates and updates up to *n* (coded as 0 through 9999999) strings in a member.
SKIPCOL(n)	Ignores *n* (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats). For NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.
MAXLEN(n)	Searches or updates up to *n* (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).
MEMBERS	Displays the names of members that contain the find string without changing the current member group.
MEMLIST	Same as ML. Any member containing the find string is selected for MEMLIST display after all REPLACE activity is complete. If no members are selected, a null sublist is the result.
ML	Same as MEMLIST. Any member containing the find string is selected for MEMLIST display after all REPLACE activity is complete. If no members are selected, a null sublist is the result.
NEWML	Same as MEMLIST and ML except that the current MEMLIST is reset.
SUBLIST	Any member containing the find string is selected for inclusion in a new sublist after all REPLACE activity is complete. If no members are selected, a null sublist is the result.
FROMKEY(ky)	For VSAM data sets only, ky is coded as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'; access begins at the first record whose key Use tatches (or is greater than) the portion of the key specified. This parameter with TOKEY. Specify it only for an alternate index or a key-sequenced data set.

FROMADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Can be specified for any VSAM data set component if control interval processing is being used ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is coded as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'; access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it for an alternate index or a key-sequenced data set.
TOADDRESS(ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not match the beginning of a logical record (or a control interval for control interval processing). If you specify this parameter for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Can be specified for any VSAM data set component if control interval processing is being used ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed

TONUMBER(nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it for a variable or fixed relative record data set.

Remarks

If you specify the WRITE or UPDATE keyword, the data is updated in-place; otherwise, only potential changes are listed.

Unprintable characters for 3270-type devices are translated to periods before they are displayed.

REPLACE formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, and DUMP. For load modules, the NUM, SNUM, and NONUM formats are equivalent to the LDUMP format and for VSAM data sets, the NUM, SNUM, and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time a format operand is entered on a REPLACE, FIND, or LIST subcommand, the value entered is used as the output format for subsequent REPLACE, FIND, and LIST subcommands. Also, note the following:

1   Formats NUM, SNUM, and NONUM limit the search and update length for logical records to 256 characters.

2   Column 72 is not searched or modified for formats NUM or SNUM if the record format is fixed with 80 character records since this is the continuation column.

3   Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is being performed). The other formats apply to logical records. For load modules, LBLOCK and LDUMP formats update and display only CSECT data.

4   Formats LBLOCK, LDUMP, BLOCK, and DUMP display only the updated segments of a record that contain the search string; however, a following segment also displays if the string spans a segment boundary.

5   Formats LBLOCK, LDUMP, BLOCK, and DUMP require equal length search and replacement strings.

6   Formats NUM, SNUM, and NONUM allow different length strings with string expansion (shifting the data over to multiple blanks) and string contraction (shifting the data up to the next blank into the replacement data). The string expansion and contraction algorithms modify programming language (Assembler, CLIST, COBOL, FORTRAN, ISPF, PL/I) statements correctly.

7   For VSAM DATA or INDEX components, the LIST, FIND, and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

Control interval access is useful if a VSAM data set has logical errors. REPLACE repairs the error; however, since only the component is opened for update, the next access of

the data set through the related cluster gets warning errors because of the differing time stamps.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
------------------------ ISPMODE Session# 1 Log# 1 --  ROW 452 TO 469 OF 473
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER004  MEM=(TAPEA  ----------------------------
>----->replace m:z /pdse221a/pdse310/ sublist
PDS246I NOWRITE is in effect; no updates will be performed

** REPLACE  TAPEA
001400 //ICNTL    DD DSN=SER07.PDSE310.CNTL,DISP=SHR
001700 //ILOAD    DD DSN=SER07.PDSE310.LOAD,DISP=SHR
PDS142I 57 lines in this member
PDS146I 2 strings found

** REPLACE  TAPEXOC
002100 //IASSM    DD DSN=SER07.PDSE310.ASSEMBLE,DISP=SHR
PDS142I 45 lines in this member
PDS146I 1 strings found

PDS147I 82 members searched
PDS148I 2 members found
PDS149I 3 total strings found

PDS165I Members are: TAPEA, TAPEXOC

PDS193I This group contains 2 members
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
------------------------ ISPMODE Session# 1 Log  ROW 1,000 TO 1,014 OF 1,014
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=PDSDECRY  -------------------------
>----->repl pdsdecry x4780c1x4770c1x
PDS246I NOWRITE is in effect; no updates will be performed

** REPLACE  PDSDECRY
PDS141I AT 000000  CSECT ENCRYPT   LENGTH 000015E0
  000180 0180  C1D9E240 5000F000  0A0612FF 4770C1A4  *ARS &.0.......Au*
  0001B0 01B0  30509180 40064770  C1D248E0 400406E0  *.&j. ...AK...
  0001E0 01E0  4770C1FC 48E04004  06E05810 400047F0  *..A..... ..0*

PDS141I AT 0017C8  CSECT R050A90     ENTRY DECIPH
  0018A8 0118  CB2C4740 C12A4770  C13259D0 CB304770  *... A...A......*
  0018B8 0128  C1324140 000247F0  C1364140 0001D237  *A.. ...0A.. ..K.*

PDS142I 7 blocks in this member
PDS146I 5 strings found
****************************** BOTTOM OF DATA ******************************
```

```
 FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,012 OF 1,012
COMMAND ===>                                               SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007 ----------------------------------
>----->repl 'cylx' 'cyly' block write
PDS140I BLOCK RECORD      6   LENGTH    80   RBA 00000400
000XX000 000 *000XX000           IF &CYLY = &STX() THEN -                 *

PDS140I BLOCK RECORD      7   LENGTH    80   RBA 00000480
000XX100 000 *000XX100              SET &CYLY = 1                         *

PDS140I BLOCK RECORD      9   LENGTH    80   RBA 00000640
000XX300 000 *000XX300              SET &ZEDSMSG = &STX(CYLY NOT NUMERIC)  *

PDS142I 80 blocks in this data set
PDS146I 3 strings found
PDS145I 3 blocks updated
****************************** BOTTOM OF DATA ******************************
```

REPRO Subcommand

The REPRO subcommand rebuilds members. It can also create new members based on existing members in the data set. REPRO cannot reblock load members.

Example
REPRO   mem1:mem2   maxblk(2400)

Syntax
```
REPRO   memgroup [ADDZAP|ADDZAP(numb)  ]
                 [ALIAS|NOALIAS        ]
                 [CALL|NOCALL          ]
                 [SUMMARY|LIST|NOLIST  ]
                 [MAXBLK(size)         ]
                 [REPLACE|NOREPLACE    ]
                 [TO(pname) | AS(pname)]
```

Aliases
REPR, REPRO

Defaults

memgroup, MAXBLK(current BLKSIZE), NOREPLACE, ALIAS for load members unless AS or TO is entered; otherwise NOALIAS.

Required

Operands

memgroup	Identifies the members to be rebuilt (or used as input for new members).
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
ADDZAP	For load members only, adds a ZAP IDR record (each ZAP IDR record contains space for 19 historical entries).
ADDZAP(numb)	For load members only, adds 1 to 99 ZAP IDR records.
ALIAS	Processes any associated members (aliases, apparent aliases, and associated main members too).
NOALIAS	Processes only specifically identified members.
CALL	Specifies that the copy program is to be invoked.
NOCALL	Specifies that the copy program is not to be invoked; member checking and data set allocations are still performed.
SUMMARY	Specifies that only summary and error messages are to be displayed.
LIST	Specifies that all messages are to be displayed.
NOLIST	Specifies that no messages are to be displayed.
MAXBLK(sz)	For RECFM=V or RECFM=F members only, specifies the maximum output block size to be used when the members are reconstructed.
REPLACE	For use with AS or TO only, replaces current members in the data set with any identically named output members.
NOREPLACE	For use with AS or TO only, generates an error message and terminates if any members in the data set will be replaced by any identically named output members.
TO(pname)	*pname* defines output member name prefixes. For example, with TO(new) and the input members, AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are created (the original members are not changed).
	If only a single member is input and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.
AS(pname)	*pname* defines output member name prefixes. For example, with AS(new) and the input members, AA, ABBA, and IBM04, members named NEW, NEWA, and NEW04 are created (the original members are not changed).
	If only a single member is input and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.

Remarks

If you use the TO (or the equivalent AS) keyword, each new member is named in a generic fashion. For example, if you enter TO(X12), each resulting member name is formed by placing X12 over the first three characters of the original member name. An exception to this rule is made when only a single member is to be created and its aliases are not also requested. In this case, the member name is taken directly from the TO keyword.

For load members, a null ZAP IDR record can be added when the member is rebuilt; for non-load members, the member can be reblocked as specified by the MAXBLK keyword.

> **NOTE** This subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection" on page 425.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
 ------------------------------------------------------------------------------
 ------------------------- ISPMODE Session# 1 Log# 1 --  ROW 145 TO 152 OF 152
 COMMAND ===>                                                 SCROLL ===> CSR
 Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.CLISTV,VOL=SER=STR92T  MEM=BC:BC6  ------------------------
 >----->repr aa* to(bc) repl
 PDS051I BC was replaced; input=1; output=1
 PDS051I BCAA was created; input=6; output=6
 PDS051I BCC was replaced; input=44; output=44
 PDS051I BC2 was created; input=1; output=1
 PDS051I BC4 was created; input=34; output=34
 PDS051I BC5 was created; input=1; output=1
 PDS051I BC6 was created; input=90; output=90
 ***************************** BOTTOM OF DATA ********************************
```

RESTORE Subcommand

The RESTORE subcommand resurrects deleted members (deleted members are physically present in a partitioned data set until the data set is compressed). Former versions of a changed member are also available for RESTORE. RESTORE is not supported for PDSE data sets.

If you use the MEMLIST (or ML) operand to restore deleted members into a MEMLIST library, several MEMLIST line commands are available to help you examine and manipulate deleted members.

Example

```
RESTORE   str   memlist   find('mydata')
```

Syntax

```
RESTORE   member   [COUNT(number)             ]
                   [DISPLAY|NODISPLAY          ]
                   [FIND('any string')         ]
```

```
              [ASIS|IGNORE                ]
              [LIKE(xyz)                  ]
              [MODULE({* | Fullm | Partm*)}]
              [PROMPT|NOPROMPT            ]
              [REPEAT|NOREPEAT|MEMLIST    ]
              [STATS|NOSTATS             ]
              [TTR(start)                 ]
```

Aliases

RES, REST, RESTO, RESTOR, RESTORE

Defaults

COUNT(5), DISPLAY, PROMPT, STATS, TTR(0)

Required

member

Operands

member	Specifies a name for the member to be resurrected. This member name becomes the default member name after the member is restored.
COUNT (num)	Displays at most *num* informational lines from a deleted member.
DISPLAY	Displays data from deleted members.
NODISPLAY	Does not display data from deleted members.
FIND('any')	Considers only deleted members containing the string ANY. If ANY is not present in a deleted member, that member is not to be displayed or resurrected.
ASIS	For use with FIND, does not translate the FIND string to upper case.
IGNORE	For use with FIND, searches for upper- and lower-case string arguments.
LIKE(xyz)	For load modules, bases the following attributes on their corresponding values from member XYZ: AMODE, authorization, editable, executable, only loadable, page aligned, reentrant, refreshable, reusable, RMODE, and SSI. For source members, base any SSI value or ISPF statistics on those present for member XYZ.

MODULE (name)	Specifies a 1- to 8-byte partial external name that limits candidates for restore processing. If the name parameter is not satisfied for any CSECT or ENTRY symbols in a deleted member, that member is not to be displayed or resurrected.The MODULE operand has several valid forms:
	■ **MODULE(*)** – uses the previous name entered on any MODULE keyword
	■ **MODULE(Fullm)** – considers only deleted members containing a CSECT or ENTRY called FULLM
	■ **MODULE(Partm*)** – considers only deleted members containing a CSECT or ENTRY called PARTM...
PROMPT	Requires a YES response to a StarTool FDM prompt before restoring a deleted member.
NOPROMPT	Does not prompt before restoring a deleted member.
REPEAT	Restores one or more deleted members. The member name to be used is taken from the partial member name entered filled to eight characters by a sequential count field. For example, if you enter RESTORE GASM REPEAT, the restored member names will be GASM0001, GASM0002, GASM0003, ...
NOREPEAT	Restores at most one member.
MEMLIST	Restores deleted members into the MEMLIST. Member names are generated from the partial member name provided and filled with a count as for the REPEAT option. These members are not actually restored; they are marked as *DELETED and placed in the MEMLIST so that you can examine them with line commands DELINK, DISASM, FIND, HISTORY, LIST, MAP, and REVIEW. Use the RESTORE line command to restore these deleted members.
STATS	For source members, creates ISPF statistics for the restored member.
NOSTATS	Does not create ISPF statistics for the deleted member.
TTR(strt)	Identifies the starting TTR address (in 1 to 6 hexadecimal digits) of the deleted member. If REPEAT/MEMLIST, DISPLAY, or PROMPT are in effect, strt is the starting TTR address for a deleted member search. If you specify TTR(0) is specified or defaulted, the member search begins at the first location following the directory.

Remarks

For maximum effectiveness in restoring deleted members use the FIND keyword to limit the number of displays and prompts. In addition, for load libraries, use the MODULE keyword to screen members by CSECT and ENTRY names. Combine these keywords with the MEMLIST option (which forces NOPROMPT) to bring deleted members into the MEMLIST only. Members are restored individually with the RESTORE line command.

The load attributes authorized, not editable, not executable, only loadable, page boundary, reentrant, refreshable, and reusable cannot be determined from load module records. These attributes can be assigned based on a model member with the LIKE operand or they can be selectively assigned by the subcommand later.

If REPEAT/MEMLIST, DISPLAY, or PROMPT are in effect, StarTool FDM searches for deleted members and changes the normal RESTORE behavior:

1 If REPEAT/MEMLIST is in effect, multiple members can be processed with a single RESTORE subcommand.

2 If DISPLAY is in effect, each deleted member is presented to you in order.

3 If PROMPT is in effect, StarTool FDM requires that you decide what to do with each deleted member.

- To restore the deleted member, reply with YES

- To not restore the deleted member, reply with NO. The next deleted member is then presented to you

- To terminate the RESTORE subcommand with no further prompting, reply with CAN

If the data set is a load library, and if the member name is an imbedded external name within the load module, then that external name is used as the entry point; otherwise, the first CSECT in the load module is used as the entry point.

It may be possible to resurrect members that were deleted before compressing a data set if you mark all of the data set's space as in-use with a FIXPDS MAXSPACE command before the restore attempt.

It may also be possible to resurrect members lost from a PDS after the directory is overwritten by sequential output due to some sort of JCL or procedural error. The sequential data blocks take one or more tracks followed by the end of file marker in the disk tracks formerly occupied by the directory. To attempt recovery of PDS members located after the overwritten tracks, type:

1 **FIXPDS MAXSPACE**
(to mark all allocated space in use)

2 FIXPDS INITDIR(43) ADDTRK(1)
(to reinitialize the directory and add another extent)

3 RESTORE name REPEAT NOPROMPT NODISPLAY
(to restore all deleted members)

4 VERIFY :
(to find invalid members after the previous DS1LSTAR)

5 BROWSE name
(to scan contents of members)

6 RENAME name realname
(to name needed members properly)

7 DELETE name
(to delete members not needed)

8 COMPRESS
(to clean up unused space again)

> **NOTE** This subcommand modifies the data set. To ensure data set integrity, allocate the data set as OLD or be aware of the data set update protection provided by the StarTool FDM command for SHR allocations. For details, see Appendix B, "Update Protection".

RESTORE bbb memlist find('vsam') produces this sample from a source library.

```
 FUNCTIONS   CONTROL A-M  LINE CMDS A-M  DSN CMDS   MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z             MEM CMDS N-Z  DEFAULTS
------------------------- Source MEMLIST 1, Session# 2 ------   ROW 1 TO 9 OF 9
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=BBB:BBB  --------------------------
 CMD  NAME     DATA/MSG VER.MOD   CREATED  LAST MODIFIED  SIZE   INIT    ID
      A                 01.00   93/04/08 93/04/12 13:21     5      5   SER07
      ABLK              01.01   93/04/09 93/04/05 14:37     6      9   SER07
      BBB00001 *DELETED 01.00   93/04/20 93/04/20 10:10    24     24   *DELETE
      BBB00002 *DELETED 01.00   93/04/20 93/04/20 10:10    12     12   *DELETE
 rest BBB00003 vsam#mem 01.00   93/04/20 93/04/20 10:10     3      3   *DELETE
      BBB00004 *DELETED 01.00   93/04/20 93/04/20 10:10    25     25   *DELETE
      LOADMARK          01.02   93/02/26 93/04/25  6:43    50     48   SER07
      SAMPSECR          01.26   90/03/27 93/04/01  5:57   197    220   SER07
      TRIAL601          01.00   93/04/14 93/04/20 15:16   123    123   SER07
***************************** BOTTOM OF DATA *******************************
```

RESTORE $$$$ memlist module(pdsmain) produces this sample from a load library.

```
 FUNCTIONS   CONTROL A-M  LINE CMDS A-M  DSN CMDS   MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z             MEM CMDS N-Z  DEFAULTS
------------------------ Load MEMLIST 1, Session# 2 -------   ROW 1 TO 4 OF 4
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=$$$$:$$$$  ------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED -- UTES  -- APF  MODE  MAIN
map  $$$$0001 *DELETED          93/04/12                          *DELETED
     $$$$0002 *DELETED          93/04/15                          *DELETED
rest $$$$0003 *DELETED          93/04/16                          *DELETED
     STARTOOL *REPRO*           93/04/16 RENT REUS
***************************** BOTTOM OF DATA *******************************
```

REVIEW Subcommand

The REVIEW subcommand browses data. Use the TSO REVIEW command. Any REVIEW operands can be added after the member name. REVIEW operates independently of ISPF.

Example
```
REVIEW   mema:memb
```

Syntax
```
REVIEW   memgroup [operands]
```

Aliases
```
REV, REVI, REVIE, REVIEW
```

Defaults

Required

Operands

memgroup	Identifies the members to be browsed.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
	If the memgroup is specified as a **:** (for all members), the REVIEW command is called to provide a member selection list.
operands	Optional. Can include any TSO REVIEW operands.

Remarks

This interface is optional. Use it only if your installation installed the public domain REVIEW command.

Recent versions of the REVIEW command operate in seven-color mode. REVIEW can display VSAM data sets. In a MEMLIST, use REV as a line command to display deleted members before they are restored.

Commands — S to U

SEPARATE Subcommand

The SEPARATE subcommand splits a member or data set containing members with ./ separators into their component members. ISPF statistics are maintained.

Example

```
SEPARATE   anymemb  output.data.set
```

Syntax

```
SEPARATE   member dsname
           [VOLUME(volser)    ]
           [SEPCHAR(dd)        ]
           [TRANSFROM(ab)      ]
           [TRANSTO(cd)        ]
           [CALL|NOCALL        ]
           [SUMMARY|LIST|NOLIST]
           [OLD|SHR | NEW [BLK|TRK|CYL]
                          [DIR(num)]  [SPACE sec)]]
                          [ROUND]   [EXPDT(yyddd)|RETPD(nnnn)]  [RLSE]
                          [STORCLAS(sclass)]   [MGMTCLAS(mclass)]
                          [DATACLAS(dclass)]   [DSNTYPE(LIBRARY|PDS)]
```

Aliases

```
SEP, SEPA, SEPAR, SEPARA, SEPARAT, SEPARATE
```

Defaults

SHR, SEPCHAR(./)

If NEW is specified, BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT, STORCLAS, MGMTCLAS, DATACLAS and DSNTYPE default to values used for the input data set.

Required

member, dsname

Operands

member	Identifies the member that contains the input for the SEPARATE function.

dsname	Identifies the output data set. Use Standard TSO syntax.
VOLUME(volser)	Identifies the volume name for output data sets that are not cataloged.
SEPCHAR(dd)	Names the delimiter to be expected between members (usually ./).
TRANSFROM(ab)	Defaults to <>. Specifies a character string that is translated to the TRANSTO string if found in column one of the input stream.
TRANSTO(ab)	Defaults to ./. Specifies a character string that is substituted for the TRANSFROM string if found in column one of the input stream.
CALL	Invoke the copy program.
NOCALL	So not invoke the copy program. Member checking and data set allocations are still performed.
SUMMARY	Display only summary and error messages.
LIST	Display all messages.
NOLIST	Do not display messages.
OLD	Obtains exclusive use of the output data set.
SHR	Allows shared use of the output data set. StarTool FDM provides output data set protection. see Appendix B, "Update Protection" on page 425.
NEW	Creates the output data set. When you specify NEW, you may also specify BLK/TRK/CYL, DIR, SPACE, ROUND, EXPDT (or RETPD), RLSE, STORCLAS, MGMTCLAS, DATACLAS, and DSNTYPE keywords.
DIR(num)	The number of directory blocks for the output data set. If DIR is not entered, the default is taken from the input data set.
SPACE(prim,sec)	The primary and secondary quantities for the output data set in track or cylinder units. If BLK, TRK, or CYL is not entered but SPACE is entered, the space allocation units are assumed the same as the current allocation.
BLK	Allocates the output data set in blocks. If BLK is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.
TRK	Allocates the output data set in tracks. If TRK is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.
CYL	Allocates the output data set in cylinder units. If CYL is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.
ROUND	Used with BLK to specify that the data set allocation be rounded up to cylinder boundaries.
EXPDT(yyddd)	The expiration date for the data set. On and after the expiration date, you can delete or write over the data set. You can also use the format EXPDT(yyyyddd).

RETPD(nnnn)	The number of days (0 to 9999) that the data set be retained by your data center.
RLSE	Release any unused space in the data set after the copy operation.
STORCLAS(sc)	The storage class for the data set. Use this parameter to identify performance and availability requirements for data sets. Use this parameter instead of UNIT and VOLUME keywords for SMS managed data sets to determine where the data set is allocated. To nullify this parameter, type **STORCLAS(*).**
MGMTCLAS(mc)	The management class for the data set. This parameter establishes the migration, backup and space release characteristics for SMS managed data sets. To nullify this parameter, type **MGMTCLAS(*).**
DATACLAS(dc)	The data class for the data set. Use this parameter to provide data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL and some VSAM SHR options for SMS managed data sets. To nullify this parameter, type DATACLAS(*). Use DATACLAS for data sets that are not SMS managed.
DSNTYPE(typ)	Specify typ as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS-managed data set.

Remarks

The SEPARATE subcommand splits apart members that were joined with a COMBINE subcommand using an IEBUPDTE (or the default) keyword.

ISPF statistics are preserved if possible and any >< control statements in a member are converted to ./ control statements.

COMBINE and SEPARATE are paired subcommands. To join members into a combined member or data set, use the COMBINE subcommand. To split the members apart again, use the SEPARATE subcommand or IEBUPDTE if the logical record length is 80.

```
FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session# 1 Log# 1 --- Row 547 to 557 of 557
 COMMAND ===>                                              SCROLL ===> CSR
 - DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=(#MDSECT ------------------------
 >------>separate download lib.pdse
 PDS290I SEPARATE is in progress
 PDS051I #MDSECT was replaced; input=1,657; output=1,656
 PDS051I @FILEXID was replaced; input=5,853; output=5,852
 PDS051I @FILEPAR was replaced; input=2,265; output=2,264
 PDS051I @SERVICE was replaced; input=5,686; output=5,685
 PDS051I @VERIFY was replaced; input=1,954; output=1,953
 PDS051I PDS0520 was created; input=473; output=472

 PDS054I Totals -- Members=6; Input=17,882; Characters=1,430,560

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

SMPGEN Subcommand

The SMPGEN subcommand assists in creating SMP/E sysmods. While only ++MAC, ++MACUPD, ++MOD (with appropriate ++JCLIN), ++SRC, ++SRCUPD statements are generated, complete SMP/E functions with an editor by using SMPGEN output as a basis for the sysmod.

In each case, begin with a library; MOD supports source libraries (assumed to be object code) and load libraries (assumed to be the final form of each load module). Choose one of the primary operand types (MAC, MACUPD, MOD, SRC, or SRCUPD) and other operands as appropriate.

To generate a generic data type, use the MAC and TYPE keywords as in following example.

Example

```
SMPGEN  : MAC distlib(apdspan) syslmod(pdspan) relfile(3) type(panel)
```

Syntax

```
SMPGEN  memgroup
        MAC    DISTLIB(ddname) [SYSLIB(ddname)]
               [SSI(hexdata)]  [TYPE(name)]
               TXLIB(ddname) | RELFILE(number) | INLINE

        MACUPD DISTLIB(ddname) [SYSLIB(ddname)]

        MOD    DISTLIB(ddname) [SYSLMOD(name) | JCLIN(member)]
               LINK|COPY] [CSLIST]
               TXLIB(ddname) | RELFILE(number)
                  | LKLIB(ddname) | INLINE [CSECT]

        SRC    DISTLIB(ddname) [SYSLIB(ddname)]
               [DISTMOD(ddname)] [SSI(hexdata)]
               TXLIB(ddname) | RELFILE(number) | INLINE

        SRCUPD DISTLIB(ddname) [SYSLIB(ddname)] [DISTMOD(ddname)]
```

Aliases

```
SM, SMP, SMPG, SMPGE, SMPGEN
```

Defaults

memgroup, TYPE(MAC)

Required

DISTLIB and either MAC, MACUPD, MOD, SRC, or SRCUPD

Operands

memgroup	Identifies the group of members for which SMP/E control statements are wanted. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
MAC	Defines MACRO elements with ++MAC statements. Data is included after each MAC statement if INLINE is specified; otherwise, either TXLIB or RELFILE is required.
MACUPD	Defines MACRO element updates with ++MACUPD statements. Data is included inline after each MACUPD statement. SMP/E assumes input data has valid IEBUPDTE sequence numbers.
MOD	Defines load or object elements with ++MOD statements. In a load library, JCLIN data is generated and a SYSLMOD operand is required. The INLINE keyword is supported for load libraries; otherwise, enter TXLIB, LKLIB, or RELFILE. In an object library, specify the name of a member that contains SMP/E JCLIN for inclusion in the generated data with the JCLIN operand. Choose to include each object deck after the MOD statement by specifying INLINE or you may use TXLIB, LKLIB, or RELFILE.
SRC	Defines source elements with ++SRC statements. Data is included after each SRC statement if INLINE is specified; otherwise, TXLIB or RELFILE is required.
SRCUPD	Defines source element updates with ++SRCUPD statements. Data is included inline after each SRCUPD statement. SMP/E assumes input data has valid IEBUPDTE sequence numbers.
DISTLIB (dd)	Adds DISTLIB(dd) to each generated SMP/E control statement to specify the DDNAME of the distribution library. The DISTLIB operand is always required for a SMPGEN subcommand.
SYSLIB(dd)	Adds SYSLIB(dd) to each generated SMP/E control statement to specify the DDNAME of the target system library. The SYSLIB operand is always optional for a SMPGEN subcommand.
SSI (hexdata)	Adds SSI(hexdata) to each generated SMP/E control statement to specify system status information. The SSI operand is always optional for a SMPGEN subcommand. The SSI operand is ignored by SMP/E unless data is included inline.
TXLIB(dd)	Adds TXLIB(dd) to each generated SMP/E control statement to specify the DDNAME where MAC, MOD or SRC elements reside.
RELFILE (num)	Adds RELFILE(num) to each generated SMP/E control statement to specify the relative number of the SMP/E relative file where MAC, MOD, or SRC elements reside.
LKLIB(dd)	Adds LKLIB(num) to each generated SMP/E control statement to specify the DDNAME where the load module MOD elements reside.
INLINE	Generates MAC, MOD, or SRC elements inline.
CSECT	Use the CSECT keyword with INLINE and MOD for a load library. ++MOD statements are placed in the delinked input on CSECT boundaries.
LINK	Default for MOD. Generates linkage-editor control statements.
COPY	Optional for MOD. Generates IEBCOPY control statements.

CSLIST	Optional for MOD. Generates all CSECT names with a CSECT keyword.
SYSLMOD (name)	Adds the specified name in the generated JCLIN for each SYSLMOD statement. This operand is required for load module libraries. As an example, if SYSLMOD(PDSLOAD) were specified, SYSLMOD statements similar to the following are generated: `//SYSLMOD DD DISP=SHR,DSN=SYS1.PDSLOAD`
JCLIN (member)	Adds the specified member contents to the generated SMP/E control statements after a ++JCLIN statement. This operand is not allowed for load module libraries and it is optional for object libraries—the member must be located in the same library as your object modules.
DISTMOD (dd)	Adds DISTMOD(dd) to each generated SMP/E control statement to specify the DDNAME of the link edit distribution library for object code produced from the assembly of source code.
TYPE (name)	Changes generated SMP/E control statements from ++MAC to ++name. For example, to generate ++PANEL statements, code TYPE(PANEL).

Remarks

The SMPGEN subcommand assists you in preparing SMP/E SYSMODs. It creates the basic framework of the required individual elements. You add other SMP/E control statements to create a valid FUNCTION, APAR, PTF or USERMOD.

The SMPGEN subcommand generates JCLIN statements for load modules or MALIAS statements for ++MAC elements.

Many valid SMP/E keywords are not supported for the different element types. Generally, only those keywords that apply to an entire library are supported. If a specific operand is required for a specific member, choose to change the generated statements after the SMPGEN process with an editor. The following SMP/E keywords are not supported:

MAC	ASSEM, DELETE, DISTMOD, DISTSRC, PREFIX, RMID, UMID, VERSION
MACUPD	ASSEM, DISTMOD, DISTSRC, MALIAS, PREFIX, VERSION
MOD	DALIAS, DELETE, LEPARM, LMOD, RMID, TALIAS, UMID, VERSION
SRC	DELETE, RMID, UMID, VERSION
SRCUPD	VERSION

SMPGEN operates as follows:

1 The data set directory is processed to include any aliases of the member group to be processed.

2 Error messages are issued before any output is generated to assist you in correcting errors.

3 ++MAC, ++MACUPD, ++MOD, ++SRC, or ++SRCUPD SMP/E statements are generated as appropriate and some additional processing is performed depending on the element type:

■ ++MAC elements have MALIAS operands added that reflect the relationship of the elements in the SMPGEN library. If INLINE is specified, MACROS are generated inline.

- MACUPD elements are always generated inline. Alias members are ignored by StarTool FDM.

- MOD elements from an object library are generated inline if the INLINE keyword is in effect. If the JCLIN operand is specified, a ++JCLIN statement is generated just before displaying the data from the JCLIN member specified.

    ++MOD elements from an load library also are generated inline using a delinked load module. The load modules are examined to generate appropriate JCLIN statements.

    For load and object libraries, ++MOD statements are not generated for alias members since they are defined to SMP/E through the JCLIN.

- ++SRC elements are generated inline if the INLINE keyword is in effect. Alias members are not allowed.

- ++SRCUPD elements are always generated inline. Alias members are ignored by StarTool FDM.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------- ISPMODE Session# 2 Log  ROW 1,000 TO 1,025 OF 1,086
COMMAND ===>                                             SCROLL ===> CSR
- DSN=SER07.PDSE311.PANELS,VOL=SER=SER006  MEM=PDSZ*  -----------------------
>----->smpgen pdsz* mac distlib(panlib) type(panel) relfile(3)
++PANEL(PDSZA@C ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZA@P ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZABA ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZABE ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZAFTZ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZAID ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZALIA) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZALLO) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZALOC) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZAOF ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZATB ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZATTR) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZATU ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZAUTH) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZAZA ) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZBROW) DISTLIB(PANLIB) RELFILE(3)  .
++PANEL(PDSZCREA) DISTLIB(PANLIB) RELFILE(3)  .
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ------------  ROW 15 OF 94
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.PDSE.ASMBLR,VOL=SER=STR804  MEM=PDS#S* -----------------------
>----->smpgen pds#s* mac inline distlib(pdsasmb)
++MAC(PDS#SIZE) DISTLIB(PDSASMB)
   .
         MACRO
&MD      PDS#SIZE &SPFSIZE=1000,  /*  SIZE OF DIALOG DISPLAY TABLE   */X
                  &SPFMAX=2000,    /*  MAXIMUM LINES FOR A SUBCOMMAND */X
                  &SPFCKPT=250,    /*  LINES BETWEEN CHECKPOINTS      */X
                  &TRPSIZE=3000    /*  MAXIMUM LINES TO TRAP          */
         GBLA  &SIZECNT
         GBLC  &SIZES(4)
&SIZES(1) SETC '&SPFSIZE'
&SIZES(2) SETC '&SPFMAX'
&SIZES(3) SETC '&SPFCKPT'
&SIZES(4) SETC '&TRPSIZE'
&SIZECNT SETA  4
         MEND
++MAC(PDS#SOPT) DISTLIB(PDSASMB)
   .
         MACRO
&MD      PDS#SOPT &LKEDDAT=Y,    /*  DEFAULT LKEDDATE IN MEMLIST?   */X
                  &SPFTDEL=Y,     /*  DROP DELETED ELEMENTS-MEMLIST? */X
                  &SPFTOLD=Y,     /*  DROP RENAMED ELEMENTS-MEMLIST? */X
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display -----------  ROW 87 OF 163
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=PDS9* --------------------------
>----->smpgen pds9* mod syslmo(pdsload) distlib(apdsload) lklib(pdsload)
++MOD(PDS98   ) DISTLIB(APDSLOAD) LKLIB(PDSLOAD)
   .
++MOD(PDS99A  ) DISTLIB(APDSLOAD) LKLIB(PDSLOAD)
   .
++MOD(PDS99HL ) DISTLIB(APDSLOAD) LKLIB(PDSLOAD)
   .
++JCLIN  .
//LKED   EXEC  PGM=IEWL,
//       PARM='NCAL,MAP,LIST,LET,RENT,REUS,REFR'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=SYS1.APDSLOAD
//SYSLMOD  DD  DISP=SHR,DSN=SYS1.PDSLOAD
//SYSLIN   DD  *
 INCLUDE APDSLOAD(PDS98)
 ORDER   PEMMAIN(P),PEMPDSIN,PEMATTNX,PEMSTAEX,PEMALLOC,PEMEXCP
 ORDER   PEMMSGS,PEMPARSE,PEMPARSX,PEMOPTIO,PEMDATEX,PEMDSNX,PEMMEMBR
 ORDER   PEXMAIN,PEXALIAS,PEXATTR,PEXBROWS,PEXCOMPA,PEXCOMPR,PEXCONT
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------- ISPMODE Session Display ---------- ROW 160 OF 200
COMMAND ===>                                            SCROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR806  MEM=@DIA* ---------------------------
>----->smpgen @dia* src distlib(pdselib) relfile(003)
++SRC(@DIABUG ) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIACAX ) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIACLN ) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIAFIND) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIAFUN ) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIAINIT) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIALALV) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIALINE) DISTLIB(PDSELIB) RELFILE(003)
  .
++SRC(@DIAOUT ) DISTLIB(PDSELIB) RELFILE(003)
  .
```

SUBLIST Subcommand

The SUBLIST subcommand forms lists (or sublists) of members. When SUBLIST gains control, it forms a new default member group using the individual member names passed to it; each subcommand that invokes SUBLIST defines a new member sublist.

For example, if a data set contains members MEMABC, MEMXYZ, MEM33 and MEZ4, entering the command SUBLIST MEM* changes the PDS300A message MEM= keyword to MEM=(MEMABC. The default member group is the list of individual member names MEMABC, MEMXYZ, MEM33 and MEZ4 instead of the rule, MEM*, that is usual with StarTool FDM subcommands.

Example
```
IF amx/   noalias  then(sublist)
    PRINT  *  form(dx00)  nohead
    COPY   *  other.library  alias
    DELETE *  alias
```

Syntax
```
SUBLIST   memgroup [ALIAS|NOALIAS      ]
                   [REVERSE            ]
                   [EXCLUDE(mem*name)  ]
                   [ONLYALIAS|ONLYMAIN ]
                   [NOBLDL             ]
```

Aliases

`SUBL, SUBLI, SUBLIS, SUBLIST`

Defaults

memgroup, NOALIAS

Required

Operands

memgroup	Identifies the group of members whose names are to be defined as a list of members.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
ALIAS	Also adds any associated members (aliases, apparent aliases, and associated main members) to the sublist of members.
NOALIAS	Adds only specifically identified member names to the sublist of members.
REVERSE	Inverts the member list to include members from the data set but not in the member list.
EXCLUDE (mem*name)	Drops members matching the combination name from the sublist.
ONLYALIAS	Forces ALIAS on and adds associated members (aliases, apparent aliases, and associated main members) to the sublist of members. Then, the main members are dropped from the sublist leaving only aliases.
ONLYMAIN	Forces ALIAS on and adds associated members (aliases, apparent aliases, and associated main members) to the sublist of members. Then, the alias members are dropped from the sublist leaving only main members.
NOBLDL	Specifies that the existence of each member in the current group need not be verified with a BLDL macro.
	This parameter is intended for high performance applications in conjunction with the MEMBERS subcommand.
	This keyword permits processing a list of members without discarding members that are not present.

Remarks

The SUBLIST subcommand disregards duplicate member names.

If multiple operations are to be performed to the same list of members, they can be organized as a single member group using the SUBLIST subcommand.

SUBLIST is useful for the THEN or ELSE keywords of the IF and FIND subcommands. Operands are not supported after the subcommand name (in the THEN or ELSE operand). However, if you use SUBLIST as the subcommand, a following subcommand (which uses an ***** for the group name) can have any operands.

SUBLIST is also useful for the THEN or ELSE keywords of the IF and FIND subcommands for situations where a refined sublist is being built. For example, to print all members named A11...XM or MEZ... with ISPF statistics which were saved in the last year and do not contain the character string NOTME, the following StarTool FDM subcommands suffice:

```
IF (ALL*xm,mez*) YEAR THEN(sublist)
    FIND * 'notme' ELSE(sublist)
    PRINT *
```

```
-------------------------- ISPMODE Session# 1 Log   ROW 1,000 TO 1,012 OF 1,012
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(PDSLA@A  --------------------------
>----->if pds* last(7) then(subl)

PDS147I 1,812 members searched
PDS148I 41 members found

PDS165I Members are: PDSLA@A, PDSLA@L, PDSLC@A, PDSLC@L, PDSLCMK5, PDSLL@A,
          PDSLO@A, PDSLOUX, PDSLT@A, PDSLT@L, PDSOPT31, PDSPNB6, PDSPNB7,
          PDSPNC4, PDSPN18, PDSSETE, PDSSETX, PDSXCP01, PDSXCP02, PDSXCP03,
          PDSXCP04, PDSXCP05, PDSXCP06, PDSXCP07, PDSXCP08, PDSXCP09, PDSXCP10,
          PDSXCP11, PDSXCP12, PDSXCP13, PDSXCP14, PDSXCP15, PDSXCP16, PDSXCP17,
          PDSXCP18, PDSXCP19, PDSXCP20, PDS4LAL, PDS4LCL, PDS4WOL, PDS9PAN

PDS193I This group contains 41 members
****************************** BOTTOM OF DATA ******************************
```

SUBMIT Subcommand

The SUBMIT subcommand submits a JCL member. Use the TSO SUBMIT command. Specify any SUBMIT operands after the member name.

Example
```
SUBMIT   mema:memb
```

Syntax
```
SUBMIT   memgroup [operands]
```

Aliases
```
SU, SUB, SUBM, SUBMI, SUBMIT
```

Defaults

memgroup

Required

Operands

memgroup	Identifies the members to be submitted for background processing. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
operands	Optional. May include any desired TSO SUBMIT operands.

Remarks

This interface is optional. Use it only as an interface to the TSO Submit command.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
------------------------- ISPMODE Session Display --------------  ROW 1 OF 6
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=PDSET*  --------------------------
>----->sub pdsback
>----->sub pdset*
PDS165I Members are: PDSETT1A, PDSETT1D, PDSETT1I, PDSETT1M, PDSETT1P,
        PDSETT1S

PDS395A Should these members be submitted (Yes/No) ?
>----->y
***************************** BOTTOM OF DATA  *******************************
```

STATUS Function

The STATUS command displays the status of the ISPMODE functions associated with the CALC, CAX, CMDTBL, CSECTS, HEX, GO, LISTA (or DDNAME), LISTC (or LISTF), LISTV, LOG, MEMLIST, PBROWSE, PEDIT, WORKPAD, and ZAP functions.

Example
STATUS

Syntax
STATUS

Aliases
ST, STA, STAT, STATU, STATUS

Operands

No operands are supported for the STATUS command.

Remarks

The STATUS display shows the status of all current ISPMODE functions in response to a STATUS command or a STATUS response from the END confirmation panel.

In StarTool FDM each function is available concurrently. This means that as you create functions (log tables, MEMLIST tables, LISTA/DDNAME tables, LISTC/LISTF tables, and so on) you can go from one to another without exiting the current function. When you select a function, you are placed back in it automatically. For example, in the log, you are placed at the top of the log for the last command. If, however, you had previously repositioned the log and you select it again with no log output additions, StarTool FDM places you at the same point again. With table panels, StarTool FDM returns you to the place at which you left the table unless one or more line commands acted on the table; in that case, positioning is for the last selected item.

A StarTool FDM function remains ACTIVE because the display table and related status information is retained even if you go to a different table. Functions that have never been selected or have been deleted are marked INACTIVE. You can activate such functions by selecting them and supplying any information required for initialization.

The final status possible is PENDING. PENDING indicates that the associated function has one or more incomplete line commands. If you select the function by entering its name, you direct the first pending line command to complete. Also, the STATUS function allows you to cancel pending line commands by function.

As noted above, one method of continuing a PENDING line command is to explicitly select the function by name. This allows the current line command to complete and initiates the next line command in that function. If no other line commands are pending, you will go to the display for the selected function.

Another way to select PENDING line commands for processing is the END command. When an END command is issued, pending functions are selected for processing using a dynamic hierarchical structure in the order shown in the STATUS function display: LOG, ZAP, CSECTS, MEMLIST, LISTA (or DDNAME), LISTC (or LISTF), LISTV, CAX, WORKPAD, CALC, HEX, CMDTBL, PBROWSE, and PEDIT. If there are no pending functions and END was issued from one of the table displays, END positions you to the log screen. If END was issued from a log screen and no line commands are pending, StarTool FDM treats the END as a request to exit the program. By default, StarTool FDM displays an END Confirmation panel and continues or terminates based on your response to the END Confirmation panel (this default can be changed in SETALL).

A final way to select PENDING line commands for processing is the PEND command. When you issue a PEND command, pending functions are selected for processing using a dynamic hierarchical structure in the order shown in the STATUS function display below. If there are no pending functions, PEND has no effect.

Line commands in one function can generate processing for another function. StarTool FDM dynamically creates a hierarchical relation between the two functions. This is best illustrated by a line command from a MEMLIST screen whose line command output is generated in the log. You are placed automatically in the log screen to review the output. You may not enter additional MEMLIST commands (other than MEMLIST with no operands) until the pending line command is completed. If you enter MEMLIST or ML with

no operands, MEMLIST completes its line command processing and dynamically removes the hierarchical relationship with the log.

```
 FUNCTIONS  CONTROL  CANCEL  DEFAULTS  FEATURES
------------------------- StarTool Function Status ------------------------
OPTION  ===>
- DSN=SER07.SMSTEST.DATA9040,VOL=SER=MVSD4B  MEM=:  ------------------------
Enter an ISPF command, a StarTool subcommand, a menu code
or choose one of the following options:
                                                         More:     +

  Option       Function    -- Description --     ---- Status -----
    8        - LOG         Session log           ACTIVE
    ZA       - ZAP         Modify a CSECT        INACTIVE
    CS       - CSECTS      CSECTs of a module    INACTIVE
    ML       - MEMLIST     Member list           ACTIVE
    LA/DD    - LISTA/DDNAME Allocation list      ACTIVE
    LC/LF    - LISTC/LISTF Catalog/file list     PENDING ACTION
    LV       - LISTV       Volume list           INACTIVE
    CAX      - CAX         Active catalog list   INACTIVE
  WORKPAD  - WORKPAD     TSO command table     INACTIVE
    CALC     - CALC        Floating calculator   INACTIVE
    HEX      - HEX         HEX calculator        INACTIVE
    CM       - CMDTBL      ISPF command table    INACTIVE
    PB       - PBROWSE     Parallel browse       INACTIVE
    PE       - PEDIT       Parallel VSAM edit    INACTIVE

Note:  Functions are shown with the lowest logical function first.
```

SVCMAP Subcommand

The SVCMAP subcommand formats information regarding the SVC table. SVCs are extensions to the operating system that are invoked to perform system tasks. SVCs receive control in supervisor state and in the master storage protect key (key zero). They have the power to alter otherwise protected storage areas or issue privileged instructions that cannot be issued directly from problem programs.

If no SVC number is requested, a formatted list of all 256 SVCs is provided. If an SVC number is specified, only that SVC is formatted. If the SVC is an ESR (router) the ESR entries are all formatted unless a particular ESR entry is requested.

Type 1 SVCs cannot link to other routines and cannot issue other SVCs from within the routine. They receive control with the local lock held and can obtain all other locks. They reside in the system nucleus (IEANUC0x).

Type 2 SVCs can link to other routines and can issue other SVCs themselves. They require no locks but can issue them. Like type 1 SVCs, they reside in the system nucleus.

Type 3/4 (3 or 4) SVCs can link to other routines and can issue other SVCs. They require no locks but can issue them. They reside in the Link Pack Area (LPA) or Modified Link Pack Area (MLPA). They can be loaded from any of the LPA libraries or alternatively, from SYS1.SVCLIB or SYS1.LINKLIB.

Type 6 SVCs cannot link to other routines or issue other SVCs. They execute disabled, meaning that no other program can gain control during the execution of the SVC. Locks cannot be held on entry to the SVC. Like types 1 and 2, type 6 SVCs reside in the system nucleus.

Four SVCs are ESR Router SVCs:

- SVC 109 contains a SVCTABLE for type 3 or 4 ESR SVCs

- SVC 116 contains a SVCTABLE for type 1 ESR SVCs

- SVC 122 contains a SVCTABLE for type 2 ESR SVCs

- SVC 137 contains a SVCTABLE for type 6 ESR SVCs

ESR SVCs are identical in format to the SVC table except for the first eight bytes. The first 3 bytes contain ESR followed by a literal 1, 2, 3, or 6 depending on which SVC type it contains. The second 4 bytes contain a binary number indicating how many ESR SVCs it contains.

If the module name output by SVCMAP is ???, this indicates that the module could not be located in the nucleus, LPA or MLPA. This is usually because the SVC was dynamically added to the SVC table by an authorized program after the system IPL. SVCMAP dumps only the DEFAULT length of these routines since the length is unknown too.

Example

```
SVC   109  ESR(11)  DISASM
```

Syntax

```
SVCMAP    [number]  [ESR(entry)    ]
                    [DUMP          ]
                    [DISASM        ]
                    [CHANGE        ]
                    [DEFAULT(size) ]
                    [MAXLEN(size)  ]
```

Aliases

```
SV, SVC, SVCM, SVCMA, SVCMAP
```

Defaults

DEFAULT(256)

Required

Operands

number	Optional. The SVC to be investigated. If no number is specified, all 256 SVCs are listed.
ESR(entry)	The ESR entry number to be investigated. If this operand is left off on an ESR SVC, all ESR entries for the SVC are listed. This operand only has meaning on an ESR SVC.

DUMP	Dump the SVC module. If the module cannot be located in the nucleus, LPA or MLPA (probably dynamically loaded), only the DEFAULT length of the module is to be dumped.
	The dump begins at the offset of the module indicated by the address from the SVCTABLE entry. The length is calculated to be the rest of the module regardless of other entry points.
	The format of the dump is 6 or 8 bytes of storage address, 6 bytes of hexadecimal offset into the module, followed by 16 hexadecimal bytes of the module at that offset, and the character equivalent surrounded by asterisks.
DISASM	Disassemble the SVC module. If the module cannot be located in the nucleus, LPA or MLPA (probably dynamically loaded), only the DEFAULT length of the module is to be formatted.
	The display begins at the offset of the module indicated by the address from the SVCTABLE entry. The length is calculated to be the rest of the module regardless of other entry points.
	The format of the display is 6 or 8 bytes of storage address, 6 bytes of hexadecimal offset into the module, followed by the operation code, reconstructed operands, the hexadecimal bytes of the module at that offset, and the character equivalent surrounded by asterisks.
CHANGE	Invoke the FINDMOD routine to locate the SVC module in the nucleus, LINKLIST or LPALIB concatenation and if found, CHANGE to that data set.
DEFAULT(Sz)	Format the length with DUMP or DISASM for a module that is not found in the in-storage nucleus, LPA or MLPA. Code Sz as a number between 8 and 99999.
MAXLEN(Mx)	The maximum length to format with DUMP or DISASM. Code Mx as a number between 8 and 99999.

```
 FUNCTIONS   CONTROL   DSN CMDS   MEM CMDS A-M   MEM CMDS N-Z   DEFAULTS   FEATURES
------------------------ ISPMODE Session# 1 Log# 1    ROW 576 TO 593 OF 1,076
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CLIST,VOL=SER=SER002  MEM=AD:LOG  --------------------------
>----->svc 109 esr(7) disasm
PDS093I ADDRESS   MODULE   LENGTH TYPE APF ESR NP AS AR LOCKS AMODE DESCRIPTION
ESR(7)  00DA5000 IGX00007 002A18 3/4  APF                       31

PDS095I IGX00007 Disassembly, length=10776
   DA5000  000000    BALR  R15,R0                  05F0            *.0*
   DA5002  000002    B     28(,R15)                47F0 F01C       *.00.*
   DA5006  000006    OR    R12,R9                  16C9            *.I*
   DA5008  000008    DC    C'GX00007 '             C7E7F0F0F0F0F740 *GX00007 *
   DA5010  000010    DC    C'91170 JR'             F9F1F1F7F040D1D9 *91170 JR*
   DA5018  000018    DC    C'M4422',X'00'          D4F4F4F2F200    *M4422.*
   DA501E  00001E    BALR  R8,R0                   0580            *..*
   DA5020  000020    LA    R10,4095(,R8)           41A0 8FFF       *....*
   DA5024  000024    LR    R12,R15                 18CF            *..*
   DA5026  000026    LR    R11,R1                  18B1            *..*
   DA5028  000028    LR    R4,R0                   1840            *. *
```

```
  FUNCTIONS   CONTROL   DSN CMDS   MEM CMDS A-M   MEM CMDS N-Z   DEFAULTS   FEATURES
------------------------- ISPMODE Session# 2 Log   ROW 1,000 TO 1,025 OF 1,257
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=SER07.PDSE311.PANELS,VOL=SER=SER006   MEM=PDSZ*  ------------------------
>----->svcmap
PDS093I ADDRESS   MODULE   LENGTH TYPE APF ESR NP AS AR LOCKS AMODE DESCRIPTION
SVC   0 00FEE898 IECVEXCP 001EB8 1                       L     24    EXCP
SVC   1 00FE22E6 IEAVEWAT 000F20 1                       L     31    WAIT
SVC   2 00FF4BA0 IEAVEPST 002954 1                       L     31    POST
SVC   3 01076388 IGC003   000D28 1                 AR    L     31    EXIT
SVC   4 011C9222 IGVVSM24 001218 1                       L     31    GETMAIN
SVC   5 011C9222 IGVVSM24 001218 1                       L     31    FREEMAIN
SVC   6 011EB198 CSVLINK  0001E0 2                       L     31    LINK
SVC   7 011EBF60 CSVXCTL  000408 2                       L     31    XCTL
SVC   8 011EB378 CSVLOAD  000188 2                       L     31    LOAD
SVC   9 011EB070 CSVDELET 000128 2                       L     31    DELETE
SVC  10 011C9E52 IGVVSM24 001218 1                       L     31    FREEMAIN
SVC  11 0280BC48 IGC0001A 0003B8 3/4                           31    TIME
SVC  12 011A95C8 CSVSYNCH 000598 2                 AR    L     31    SYNCH
SVC  13 0221F000 IGC0101C 015830 3/4            AS AR    L     31    ABEND
SVC  14 01FD4020 IEAVTESP 001BE0 3/4                     L     31    SPIE
SVC  15 0101D91A IECVPST  001C4C 1                       L     31    ERREXCP
SVC  16 010F3140 IOSPURGA 001768 2                             31    PURGE
SVC  17 02A49000 IGC0001G 0001C8 3/4                           31    RESTORE
SVC  18 00B87818 ???      000000 2                             31    BLDL
SVC  19 00E02138 IGC0001I 00DC20 3/4                           24    OPEN
SVC  20 00B8C660 ???      000000 3/4                           24    CLOSE
```

TSO Subcommand

Use the TSO subcommand to invoke a TSO command processor or CLIST. Specify any operands after the name specified.

Example

```
TSO   listc  lev(sys2)
```

Syntax

```
TSO   command [operands]
```

Aliases

```
T, TS, TSO
```

Defaults

Required

command

Operands

command	CLIST or command processor to invoke. If you use the form %cnam, CNAM is assumed to be a CLIST name.
operands	Optional. Can include any operand.

Remarks

Use the TSO subcommand to invoke a TSO command processor or CLIST. Specify any operand after the name specified.

This is the ISPF TSO command, if you enter TSO, when in ISPMODE or MEMLIST. To use the StarTool FDM TSO subcommand, enter an alias subcommand name such as T or TS.

Your installation may have installed the TSO subcommand to obtain the equivalent of "TSO TSOEXEC command operands". If this is the case, invoke authorized commands with fewer keystrokes. To determine if your installation has this option enabled, type a CONTROL DEFAULT subcommand and look for an output line beginning "TSO calls". A value of TSOEXEC indicates it is enabled.

TSOEDIT Subcommand

The TSOEDIT subcommand edits a member; the TSO EDIT command is used.

Example

```
TSOEDIT    mema:memb  cntl
```

Syntax

```
TSOEDIT    memgroup [ASIS         ]
                    [OLD|NEW      ]
                    [NONUM        ]
                    [ASM|BASIC|CLIST|CNTL|COBOL|DATA|
                     FORTE|FORTG|FORTGI|FORTH|GOFORT|
                     IPLI|LIST|PLI|PLIF|TEXT|VSBASIC ]
```

Aliases

```
TSOE, TSOED, TSOEDI, TSOEDIT
```

Defaults

memgroup, EDIT type based on the data set name

Required

Operands

memgroup	Identifies the members to be edited.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
ASIS	Edits the member with upper- and lower-case characters.
OLD	Verifies that the member exists before calling EDIT.
NEW	Verifies that the member does not exist before calling EDIT.
NONUM	Edits the member without using line numbers.
ASM	EDIT type is assembly.
BASIC	EDIT type is BASIC.
CLIST	EDIT type is CLIST.
CNTL	EDIT type is CNTL.
COBOL	EDIT type is COBOL.
DATA	EDIT type is DATA.
FORTE	EDIT type is FORTRAN level E.
FORTG	EDIT type is FORTRAN level G.
FORTGI	EDIT type is FORTRAN level GI.
FORTH	EDIT type is FORTRAN level H
GOFORT	EDIT type is GOFORT.
IPLI	EDIT type is IPLI.
LIST	EDIT type is LIST.
PLI	EDIT type is PLI.
PLIF	EDIT type is PLI (F-level).
TEXT	EDIT type is TEXT.
VSBASIC	EDIT type is VSBASIC.

Remarks

If you enter an EDIT type keyword, that keyword is passed to EDIT as the descriptive qualifier; otherwise, a descriptive qualifier is chosen as follows:

1 If the low-level qualifier of the partitioned data set name is one of the valid descriptive qualifiers for EDIT (ASM, BASIC, CLIST, CNTL, COBOL, DATA, IPLI, LIST, PLI, TEXT, or VSBASIC), that qualifier is passed to EDIT.

2 If the low-level qualifier is FORT, GOFORT is passed to EDIT. Otherwise, the data type qualifier is not a valid descriptive qualifier and the general descriptive qualifier, DATA, is passed to EDIT.

This interface is optional. Use it only as an interface to the TSO EDIT command. To terminate this interface, type END from the EDIT prompt.

TSOLIST Subcommand

The TSOLIST subcommand lists a member. The TSO LIST command is used. Specify any LIST operand after the member name.

Remarks

This interface is optional. Use it only if your installation installed the public domain LIST command.

Example

```
TSOLIST    mema:memb
```

Syntax

```
TSOLIST    memgroup   [BYPASS(numtoskip)    ]
                      [COL(begincol:endcol) ]
                      [HEX|X                ]
                      [MAX(maxtolist)       ]
                      [NUM|SNUM|NONUM        ]
```

Aliases

```
TSOL, TSOLI, TSOLIS, TSOLIST
```

Defaults

memgroup, NUM (if a non-numeric is encountered, then NONUM)

Required

Operands

memgroup	Identifies the members to be listed.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".

others	Optional. Can include any desired TSO LIST operands.

USAGE Subcommand

The USAGE subcommand displays statistics on the current data set. The DSNAME subcommand is issued automatically with this subcommand.

Example

USAGE

Syntax

USAGE   [ALL]

Aliases

U, US, USA, USAG, USAGE

Defaults

Required

Operands

ALL	Displays data set extent information and formats the DSCB fields. Any operand on the USAGE subcommand is sufficient to get all of the available data.

Remarks

The statistics displayed include:

- The number of disk storage tracks allocated the number of disk storage tracks currently in use and free

- The number of disk storage extents required for the data set

- The number of directory blocks allocated

- The number of directory blocks currently in use and free

- The number of disk storage tracks used by the directory

- The number of members in the directory

- The number of member names that are alias names

The USAGE command displays statistics for up to 59 volumes. Up to nine volumes are listed in the first panel. If more volumes are allocated, the volume list shows a plus (+) character after the eighth volume on the first panel.

Following are some sample screens of the USAGE subcommand. USAGE with no operand returns data from the first screen and the remaining screens are added for any operand after USAGE.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ---------- ROW 389 OF 452
COMMAND ===>                                               SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM ------------------------
>----->u all
PDS200I DISP UNIT    RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380    U         0   32760   2X   352      80    80 TRK       26

PDS180I Data set: CREATED   EXPIRES   LAST USE  UPDATED   LASTUSE   FIRSTUSE
PDS180I          89/02/17  0/00/00  95/06/19  95/06/19 C911407   C911401T

PDS181I Extents in tracks: 272, 80

PDS182I Tracks: ALLOCATED     USED     FREE   EXTENTS   VOLUME
PDS182I                352      272       80         2   STR804

PDS183I Directory: BLOCKS     USED     FREE    TRACKS   MEMBERS   ALIASES
PDS183I                80       54       26         2       329        45
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ---------- ROW 421 OF 452
COMMAND ===>                                               SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM ------------------------
PDS184I EXTENT  UCB  LO TT-HI TT  TRACKS     LOW CCHH-HIGH CCHH     BOUNDARY
PDS184I ------  ---  ----- -----  ------  ----------- -----------  --------
PDS184I     0  46F  00.00 01.0F     272  02.F2.00.03 03.04.00.04  TRK
PDS184I     1  46F  01.10 01.5F      80  03.68.00.08 03.6D.00.0C  TRK


PDS185I Format 1 DSCB at 01B9000722:
PDS185I 0   4   8   C   10  14  18  1C  20  24  28  2C  30  34  38  3C  40  44
PDS185I C911407.LINK.LOAD                            101T..............C911407
PDS185I CFFFFFF4DCDD4DDCC444444444444444444444444444FFFE50A00503000000CFFFFFF4
PDS185I 3911407B3952B3614000000000000000000000000000101390A0190000020039114070

PDS185I  48  4C  50  54  58  5C  60  64  68  6C  70  74  78  7C  80  84  88
PDS185I CFFFF50A000800C07F050008800500024F0000F00000000006000600000000000000000
PDS185I 3911490A000F2001F800000200001FA30101022033404113 8083D0C000000000000000
```

```
------------------------ ISPMODE Session# 2 Log# 1 --  ROW 501 TO 526 OF 559
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007  ----------------------------------
>----->u
PDS200I DISP UNIT    RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY DSORG
PDS200I SHR  3380    VSAM   200   9000    1X    1       0      1 TRK VS-KSDS

PDS121I Association: DATA-----SER07.VSAM.IS.CLUSTER.DATA
PDS121I Association: INDEX----SER07.VSAM.IS.CLUSTER.INDEX
PDS121I Association: AIX------SER07.VSAM.AX.CLUSTER
PDS121I Association: CATALOG--ICFUCAT.VTSG312

PDS180I Data set: CREATED    EXPIRES    LAST USE    UPDATED   SECURITY
PDS180I          1993/01/06  **NONE**   **NONE**    NO        READ

PDS182I Tracks: ALLOCATED    USED    FREE   EXTENTS
PDS182I                 1       1       0       1

PDS195I INDEXED     SPANNED    NOIMBED    NOREPLICAT SHROPTNS(1,3)
PDS196I NOERASE    NOWRITECHK SPEED      NOREUSE    UNORDERED

PDS197I Key length: 8
PDS197I Key offset: 0
PDS197I Average LRECL: 200
PDS197I Maximum LRECL: 9,000
PDS197I Creation date: 1993.006
PDS197I Buffer space: 12,288

PDS198I DATA space usage:  TRACKS KILOBYTES    CA's   CI's   PERCENT
PDS198I Allocated space:      1      40        1     10
PDS198I High used space:      1      40        1     10    100.0
PDS198I Real used space:      1      28        1      7     70.0
PDS198I INDEX space usage: TRACKS KILOBYTES    CA's   CI's   PERCENT
PDS198I Allocated space:      1      40        1     10
PDS198I High used space:      1       4        1      1     10.0

PDS199I Records:   TOTAL   DELETED   UPDATED   INSERTED  RETRIEVED    EXCP'S
PDS199I               80      0        8        8        4549        182

PDS241I CI Space: FREESPACE   SPLITS  %SPLITS
PDS241I                10       2     20.0
PDS241I CA Space: FREESPACE   SPLITS  %SPLITS
PDS241I                10       0      0.0

PDS242I Attributes for  DATA      INDEX
PDS242I CI size:        4096      4096
PDS242I CI's per CA:      10        10
PDS242I Allocation:    TRACK     TRACK
PDS242I Primary:          1         1
PDS242I Secondary:        1         1

PDS243I Index: LEVELS   RECORDS   HI-LEVEL
PDS243I            1       1         0

PDS244I CA splits/CI  CI splits/insert  Inserts/read
PDS244I        0.0           25.0            10.0
```

```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ----------  ROW 421 OF 452
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  ------------------------

PDS186I LOC NAME     VALUE        DESCRIPTION
PDS186I --- ----     -----        -----------
PDS186I 00  DS1DSNAM C911407.LINK.LOAD
PDS186I 2C  DS1FMTID F1           FORMAT IDENTIFIER
PDS186I 2D  DS1DSSN  F0F1E35900AA DATA SET SERIAL NUMBER
PDS186I 33  DS1VOLSQ 1.           VOLUME SEQUENCE NUMBER
PDS186I 35  DS1CREDT 590030       CREATION DATE
PDS186I 38  DS1EXPDT 000000       EXPIRATION DATE
PDS186I 3B  DS1NOEPV 2.           NUMBER OF EXTENTS ON VOLUME
PDS186I 3C  DS1NOBDB 0.           NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK
PDS186I 3D          00           RESERVED (ONE BYTE)
PDS186I 3E  DS1SYSCD C3F9F1F1F4F0F740C3F9F1F1F4
PDS186I 4B  DS1REFD  5900AA       DATE LAST REFERENCED
PDS186I 4E          0000008F     RESERVED (FOUR BYTES)
PDS186I 52  DS1DSORG 0200         DATA SET ORGANIZATION
PDS186I 54  DS1RECFM C0           RECORD FORMAT
```

```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ----------  ROW 438 OF 452
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  ------------------------
PDS186I 55  DS1OPTCD 01           OPTION CODE
PDS186I 56  DS1BLKL  32,760.      BLOCK LENGTH
PDS186I 58  DS1LRECL 80.          LOGICAL RECORD LENGTH
PDS186I 5A  DS1KEYL  0.           KEY LENGTH
PDS186I 5B  DS1RKP   0.           RELATIVE KEY POSITION
PDS186I 5D  DS1DSIND 82           DATA SET INDICATORS
PDS186I 5E  DS1SCALO 80000050     SECONDARY ALLOCATION
PDS186I 62  DS1LSTAR 010F0A       TTR OF LAST USED TRACK AND BLOCK ON TRACK
PDS186I 65  DS1TRBAL 9,024.       BYTES REMAINING ON LAST TRACK USED
PDS186I 67          F100         RESERVED (TWO BYTES)
PDS186I 69  DS1EXT1  0100.02F20003.03040004  FIRST EXTENT DESCRIPTION
PDS186I 73  DS1EXT2  0101.03680008.036D000C  SECOND EXTENT DESCRIPTION
PDS186I 7D  DS1EXT3  0000.00000000.00000000  THIRD EXTENT DESCRIPTION
PDS186I 87  DS1PTRDS 0000000000   CCHHR OF ANY ASSOCIATED FORMAT 2 OR 3 DSCB

***************************** BOTTOM OF DATA  *****************************
```

Commands — V to Z

VERIFY Subcommand

The VERIFY subcommand validity checks data sets. For PDS or PDSE data sets, check the entire data set by entering a **:** in the member name position. Check an individual member or member group by entering its name in the member name position.

Example

```
VERIFY   mema:memb
```

Syntax

```
VERIFY   memgroup
         [LOAD|NOLOAD                ]
         [LKED|NOLKED               ]
         [MAXBLK(bsize)             ]
         [NAME|NONAME               ]
         [COUNT|NOCOUNT             ]
         [READ|NOREAD|INPUT|NOINPUT]
         [STATS|NOSTATS             ]
         [UPDATE|NOUPDATE           ]
         [MEMBERS | MEMLIST | ML | NEWML | SUBLIST      ]
       * [FROMKEY(key)|FROMADDRESS(add)|FROMNUMBER(num)  ]
       * [TOKEY(key)|TOADDRESS(add)|TONUMBER(num)        ]
       * [EXAMINE | NOEXAMINE | NOINDEXTEST | NODATATEST ]

*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases

```
V, VE, VER, VERI, VERIF, VERIFY
```

Abbreviations

FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS, and TNUM for TONUMBER.

Defaults

memgroup, LOAD, LKED, NAME, READ, STATS, NOUPDATE, EXAMINE

Required

Operands

memgroup	Identifies the members to be validity checked. If a **:** is entered in the member name position, the data set is validity checked; otherwise, only the specified members are checked individually. Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
LOAD	For load libraries, LOAD each member.
NOLOAD	Does not LOAD any members.
LKED	For load libraries, checks linkage editor attributes and LINKLIST LLA member status.
NOLKED	Does not perform any linkage editor checks or LINKLIST LLA member status.
MAXBLK(bsize)	Generates messages for members with blocks larger than the value bsize.
NAME	Checks member names for validity. Valid member names are from 1 to 8 characters long containing upper-case alphanumeric characters; the first character cannot be numeric.
NONAME	Does not check member names for validity.
COUNT	Outputs a short message giving input counts.
NOCOUNT	Outputs normal messages.
READ	Inputs each member (or the data set).
NOREAD	Does not input any data.
INPUT	Inputs each member (or the data set).
NOINPUT	Does not read any data.
STATS	Displays normal completion statistical messages.
NOSTATS	Does not display statistical messages. This checks a member or a data set for unusual conditions.
UPDATE	Tests member addition and deletion to determine if the PDS directory has any available space and if normal PDS operations are possible on the data set. For a VSAM data set, performs an IDCAMS VERIFY function.
NOUPDATE	Does not update the data set.
MEMBERS	Displays the names of members with warning or error messages but without changing the current member group.
MEMLIST	Same as ML. Any member with warning or error messages is selected for MEMLIST display. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.
ML	Same as MEMLIST. Any member with warning or error messages is selected for MEMLIST display. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.
NEWML	Same as MEMLIST and ML except that the current MEMLIST is reset.

SUBLIST	Any member with warning or error messages is selected for inclusion in a new sublist. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.
FROMKEY(ky)	For VSAM data sets only, ky is coded as the key of the first record to be accessed. This is a generic key and can be coded as x'hexkey'; access begins at the first record whose key matches (or is greater than) the portion of the key specified. Use this parameter with TOKEY. Specify it only specified for an alternate index or a key-sequenced data set.
FROMADDRESS (ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with TOADDRESS. FROMADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed
FROMNUMBER (nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. Use this parameter with TONUMBER. Specify it only specified for a variable or fixed relative record data set.
TOKEY(ky)	For VSAM data sets only, ky is coded as the key of the last record to be accessed. This is a generic key and can be coded as x'hexkey'**;** access ends after the first record whose key matches the portion of the key specified. Use this parameter with FROMKEY. Specify it only for an alternate index or a key-sequenced data set.
TOADDRESS (ad)	For VSAM data sets only, ad is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records are accessed in physical sequential order instead of key order. Use this parameter with FROMADDRESS. TOADDRESS(address) ■ Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components ■ Cannot be specified if the data set is being accessed through a path ■ Cannot be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed

TONUMBER (nm)	For VSAM data sets only, nm is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. Use this parameter with FROMNUMBER. Specify it only for a variable or fixed relative record data set.
EXAMINE	For VSAM data sets only, specifies that an IDCAMS EXAMINE with INDEXTEST and DATATEST are wanted.
NOEXAMINE	For VSAM data sets only, specifies that no IDCAMS EXAMINE is wanted.
NOINDEXTEST	For VSAM data sets only, specifies that an IDCAMS EXAMINE with NOINDEXTEST is wanted.
NODATATEST	For VSAM data sets only, specifies that an IDCAMS EXAMINE with NODATATEST is wanted.

Remarks

The VERIFY subcommand validity checks PDS or PDSE data sets in the following manner:

1 For VERIFY : UPDATE, a dummy member is added and deleted to determine if the data set directory is full and if normal PDS functions are possible. This is the only potential data set update performed by the VERIFY subcommand and it is not performed on a PDSE data set.

2 The data set directory is read, member names are placed into a program table in TTR sort order (member location order) and the following checks are performed:

- If NAME is in effect, member names are checked for validity

- If the data set is a LINKLIST library and LKED is in effect, the LLA status of members is checked with a system BLDL

- A message is issued if the LLA status of a member cannot be determined due to a module in a TASKLIB library or a member in a higher LINKLIST library

- A message is issued if a member directory entry and its LLA entry are not identical (they are not synchronized)

- A message is issued if a LINKLIST member is not known to LLA

- If the data set is a load library and LKED is in effect, the following illegal editor attribute combinations are checked:

- Modules with RENT and NOREUS attributes

- Modules with RMODEANY and AMODE24 or AMODEANY attributes

- Modules with OVLY and RENT, REUS, REFR, SCTR, RMODEANY, AMODE31, AMODE64, or AMODEANY attributes

- Modules with TEST and NOEDIT attributes

- Modules with REUS and SCTR attributes

- If the data set is a load library and LOAD is in effect, each member is checked for ABENDS during a LOAD operation

- Each member is checked for current usage by an ISPF EDIT session

- Member directory entries are checked for correct name order

- Each member TTR address is checked against the end of the data set

3 The program TTR table is checked for:

- Orphan members (aliases with no associated main member)
- Apparent alias members (main members whose start address matches another main member's start address)
- Alias load modules with incorrect directory pointers to associated main members
- Alias load modules whose RMODE entries or main AMODE entries do not match their base module's entries

4 If READ or INPUT is in effect, the directory and each member is read checking for:

- Permanent I/O errors
- Input block sizes exceeding the MAXBLK value (or the DCB BLKSIZE if MAXBLK is omitted)
- BLKSIZE divided by input LRECL not integral (RECFM=F)
- Input LRECL exceeding maximum DCB LRECL (RECFM=V)
- Input LRECL less than 4 bytes (RECFM=V)
- Null members
- Directory RLD/CONTROL counts that do not match the first RLD entry

5 Informational messages regarding the data set are provided.

The VERIFY subcommand validity checks sequential or *VSAM data sets* in the following manner:

1 For VSAM data sets if NOUPDATE is not specified, an IDCAMS VERIFY is performed on the data set.

2 For a KSDS or an Alternate index if NOEXAMINE is not specified, an IDCAMS EXAMINE is performed on the data set.

3 For a KSDS or an Alternate index if NOREAD is not specified, the data set is also read sequentially to determine if the index is synchronized with the data.

4 Sequential data sets are checked for current usage by an ISPF EDIT session.

5 If READ or INPUT is in effect, the data set is read checking for:

- Permanent I/O errors or VSAM logical errors
- Input block sizes exceeding the MAXBLK value (or the DCB BLKSIZE if MAXBLK is omitted)
- BLKSIZE divided by input LRECL not integral (RECFM=F)
- Input LRECL exceeding maximum DCB LRECL (RECFM=V)
- Input LRECL less than 4 bytes (RECFM=V)

6   Informational messages regarding the data set are provided.

```
>------>ch jcl.cntl
 PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
 PDS200I SHR  3380 C   FB      80   23440 1X    80      41    10 TRK        7


 PDS300A ENTER OPTION -- DSN=JCL.CNTL,VOL=SER=SER004   MEM=

 ** VERIFY   JOBC
PDS860E JOBC is an alias but no main member exists

** VERIFY   FIX
PDS510W This is a null member

PDS006I End of data set
PDS110I 17,174 logical records were input
PDS111I 332 physical blocks were input
PDS112I 23,440 characters in the largest physical block
PDS113I 4,138 characters per average physical block
PDS114I 0 tracks could be regained by compressing this data set
PDS115I 304 members were checked

PDS130I The following is a track usage map of the data set
PDS130I DDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXL....................
PDS130I ....................
PDS117I 304 members counted; cumulative size is 17,174 records and 1,373,920
         characters

PDS165I Members are: FIX, JOBC

PDS193I This group contains 2 members
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ------------   ROW 52 OF 70
COMMAND ===>                                            SCROLL ===> CSR
- DSN=C911407.SAM.LOAD,VOL=SER=STR807   MEM=(PDSWHO  --------------------------
>----->verify (pdswho,pdspgm)
** VERIFY   PDSWHO
PDS860E Member is an alias but no main member exists
PDS861E The alias directory entry notes the main entry name as SYSDSN

PDS111I 7 physical blocks were input
PDS112I 2,056 characters in the largest physical block
PDS113I 353 characters per average physical block

** VERIFY   PDSPGM
PDS111I 162 physical blocks were input
PDS112I 6,000 characters in the largest physical block
PDS113I 2,768 characters per average physical block

PDS118I 1 members RMODE24; size is 3K
PDS119I 1 members RMODEANY; size is 436K
```

```
 FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------- ISPMODE Session# 1 Log# 1 --  ROW 168 TO 186 OF 186
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.VSAM.CSI,VOL=SER=SER004  ----------------------------------------
>----->ver
 VERIFY FILE(SYS00303)

 EXAMINE NAME(SER07.VSAM.CSI)   INDEXTEST   DATATEST ERRORLIMIT(4)
IDC01700I INDEXTEST BEGINS
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC01701I DATATEST BEGINS
IDC01709I DATATEST COMPLETE - NO ERRORS DETECTED
IDC01708I 14 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 2113 RECORDS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 88 BYTES
IDC01722I 95 PERCENT FREE SPACE

PDS006I End of data set
PDS111I 2,113 physical blocks were input
PDS112I 88 characters in the largest physical block
PDS113I 24 characters per average physical block
```

```
>------>Verify  INPUT

PDS006I End of data set
PDS110I 12 logical records were input
PDS111I 1 physical blocks were input
PDS112I 960 characters in the largest physical block
PDS113I 960 characters per average physical block

PDS117I Cumulative size is 960 characters
```

VMAP Function

Use the VMAP command as a line command in LISTV for a particular volume or as a
primary command to format all extents of a volume.

Example
VMAP sysres

Syntax
VMAP volser

Aliases
VM, VMA, VMAP

Defaults

Required

volser

Operands

volser	Name of the online disk volume whose extents are to be mapped.

Remarks

VMAP constructs a full volume extent map (including all free space) in disk address (CCHH) order. The primary view for this table shows the size of each extent and the associated DSNAME.

The alternate (reached with a LEFT or RIGHT command) view of this function represents many statistics on each data set. The SORT, F (find), and REMOVE commands are available for data manipulation. In addition, use the LC line command to copy an entry for a particular data set to the LISTC/LISTF table for additional processing.

The following primary commands are supported directly for the VMAP function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" on page 37.

ALT[ERNAT]	Displays an alternate view of the VMAP table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	`APPLY linecmd`
DUA[L]	Displays a double line view of the VMAP table.	
EDITT[BL] (or ET[BL])	Enters an edit session on VMAP table data.	
F	Finds a string and positions the display start location.	
	Syntax `F anystring [ASIS]` `[ANY|DSNAME|DO|RECFM|LRECL|BLKSIZE|` `ALLOC|SEC|EXTENT|CCHH]`	
LE[FT]	Rotates through alternate views of the VMAP table. PF keys 10 and 22 are normally set to LEFT.	
L[OCATE]	Positions to a data line in sorted tables by searching the current sorted column for the specified data.	
	Syntax `LOCATE dsname|size|length|extents|alloc|cchh`	
NOR[MAL]	Displays the default view of the VMAP table.	

O[PTIONS]	Provides primary command selection for VMAP and operand syntax assistance.	
OUT[PUT]	Outputs the VMAP table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the VMAP table based on a string match.	

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`            [ANY|DSNAME|DO|RECFM|LRECL|BLKSIZE|`
	`             ALLOC|SEC|EXTENT|CCHH]`	

RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.	
RI[GHT]	Rotates through alternate views of the VMAP table. PF keys 11 and 23 are normally set to RIGHT.	
SO[RT]	Sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.	
	Syntax	`SORT [CCHH|SIZE|LENGTH|EXTENTS|ALLOC]` `     [ASCEND|DESCEND]`
X	Clears the VMAP relative to the cursor position.	
	Syntax	`X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the VMAP table; this is equivalent to X ALL	

The following line commands are supported in the VMAP function.

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
B	Changes to the data set and issues a MEMLIST all.
C	Changes to the data set.
DIRF	Tailors batch JCL to set the DIRF bit on for this volume.
INDX	Tailors batch JCL to reverse the indexed VTOC status for this volume (an OS/VTOC is switched to an indexed status and an indexed volume is switched to an OS/VTOC).
E	Changes to the data set and issues a MEMLIST all.
GO	Changes to the data set using GO processing (a number is optional).
IDC	Performs an IDCAMS LISTC and directs output to the log.
INFO	Displays information on the data set in panel format and updates data set information.
K	Kills and clears all following line commands.
LC	Copies this data set entry into the LISTC/LISTF table for further processing.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST all.

MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
REST	Restores a data set from free space on the volume. This method requires that the volume index be turned of. See INDX above. Use REST to recover a data set if these two conditions are met: - The original space allocated to the data set has not been re-used for another data set - If the original data was allocated with security options that erased the data set when it was deleted, the data set does not contain the original data even when re-allocated with the REST option.
RJCL	Tailors batch JCL to restore a data set from free space on the volume using ABSTR allocation. Use FIXPDS to change the DCB parameters after the restore. RJCL has the same limitations on recovering data sets as were listed under REST processing. Space allocation using ABSTR (absolute track address) may not be possible for volumes managed by DFSMS.
S	Selects line command (normally CHANGE, it is set by SETSEL).
U	Changes to the data set and displays USAGE.
UT	Selects the extended user line command panel.
W	Adds the data set name to the WORKPAD.
WHO	Checks for users of this data set with the WHOHAS subcommand.
X	Drops the table line.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported for the VMAP function:

==	= command. Repeats the previous line command for the range of lines. `*cm ERR- t="Block line command,==">`
CC	C command. Changes data set for each line.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	ML command. Changes and issues a MEMLIST all for each line.
OO	O command. Provides line command assistance for each line.
SS	S command. Selects each line in the range of table lines.
XX	X command. Drops the range of table lines.

```
  FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
 -----------------------------------------------------------------------------
 ------------------------------- VMAP STG00C   ----------- ROW 1 TO 17 OF 592
 COMMAND ===> right                                        SCROLL ===> CSR
 Enter an ISPF command or a special control code:
   ---------------------------------------------------------------------------
 CMD  CCHH    End CCHH EX# DATA/MSG ---------DATA SET NAME -------------------
      00000001 0001000E  1 *----29 SYS1.VTOCIX.STG00C_____
      00020000 0008000E  1 *---105 *** FORMAT 4 (VTOC) EXTENT ***_____
      00090000 000A000E  1 *----30 SYS1.VVDS.VSTG00C_____
      000B0000 000B0000  2 *-----1 UCIS033.ISPF350.ISPPROF_____
      000B0001 000B0001  1 *-----1 WTESSNT.PRD.R921.DBI.SRC_____
      000B0002 000B0003  1 *-----2 WTESSNT.OPS.MIG.LETTER.AU940488_____
      000B0004 000B0004  1 *-----1 WFDFA3NT.STAR.ALL_____
      000B0005 000B0005  1 *-----1 WFD5001.SPFTEMP1.CNTL_____
 +----------------+ 006  1 *-----1 UCIMSAVT.GL9002.GWF05.INDEX_____
 | TOTAL=    301 | 007  1 *-----1 *** FREE SPACE EXTENT ***_____
 | NVSAM=    275 | 008  1 *-----1 UCIMSAVT.DC9001.LKGB01.INDEX_____
 | ADDS =    592 | 009  1 *-----1 WFDFA7NT.WILSON.YR9293_____
 | EXT  =    478 | 00A  1 *-----1 ACWCS86.PS0080S_____
 | SPACE=  44418 | 00B  1 *-----1 WTESSNT.PRD.R921.EXP.CNT_____
 | FREE =  20931 | 00C  1 *-----1 UCIH003.EOXMBMGR.BKLSHELF_____
 +----------------+ 00D  1 *-----1 UCIS003.TRANSMIT.LOG_____
      000B000E 000B000E  1 *-----1 *** FREE SPACE EXTENT ***_____
```

When a VMAP function is invoked, the message lines have the following meanings:

TOTAL=	Total number of data sets displayed.
NVSAM=	Total number of non-VSAM data sets.
ADDS=	Total lines in the VMAP table.
EXT=	Total number of extents.
SPACE=	Total number of tracks allocated on this volume.
FREE=	Total allocated tracks free on this volume.

```
 FUNCTIONS   CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------------- VMAP STG00C   --------  ROW 114 TO 121 OF 587
COMMAND ===> exclude po not do                          SCROLL ===> CSR
Enter an ISPF command or a special control code:
   ----------------------------------------------------------------------------
CMD  CCHH     End CCHH EX# DATA/MSG ---------DATA SET NAME -------------------
     TYPE LENGTH DO RECFM LRECL BLKSI --SIZE- --FREE- USED ALLOC -SEC- RND EXT
     087C000D 087D0005   1 *-----8 *** FREE SPACE EXTENT ***_____
     FRE        8
     087B0006 087C0000   1 *----10 *** FREE SPACE EXTENT ***_____
     FRE       10
     02D10000 02D8000E   1 *---120 ACWCS04.CN7662.DATA_____
     CYL     120 ** FB     132 31680    120T    119T   0% CYL       8 YES   1
     02C90000 02D0000E   1 *---120 ACWCS04.CN7760.DATA_____
     CYL     120 ** FB     132 31680    120T    119T   0% CYL       8 YES   1
     015A0000 015A000E   1 *----15 ACWCS50.FNA970.MATRIX_____
     CYL      15 PS FB      80  6160     15T     13T  13% CYL       1 YES   1
     088E0000 0968000E   1 *--3285 ACWCS50.OBB.T.EXDETAIL_____
     CYL    3285 PS FB     388 23280   3645T      0T 100% CYL      50 YES   2
     0BE10000 0BF8000E   2 *---360 ACWCS50.OBB.T.EXDETAIL_____
     CYL     360 PS FB     388 23280   3645T      0T 100% CYL      50 YES   2
     0A570007 0A57000A   1 *-----4 ACWCS86.ISPF350.ISPPROF_____
     TRK       4 PO FB      80  6160      4T      0T 100% TRK       5 NO    1
```

```
 FUNCTIONS   CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------------------------------------------------------------
------------------------------- VMAP STG00C   ------------  ROW 1 TO 8 OF 194
COMMAND ===>                                            SCROLL ===> CSR
EXCLUDE PROCESSING DONE, ROWS DELETE =  393
   ----------------------------------------------------------------------------
CMD  CCHH     End CCHH EX# DATA/MSG ---------DATA SET NAME -------------------
     TYPE LENGTH DO RECFM LRECL BLKSI --SIZE- --FREE- USED ALLOC -SEC- RND EXT
     000B0000 000B0000   2 *-----1 UCIS033.ISPF350.ISPPROF_____
     TRK       1 PO FB      80  6160      4T      0T 100% TRK       1 NO    3
     000B0001 000B0001   1 *-----1 WTESSNT.PRD.R921.DBI.SRC_____
     TRK       1 PO FB      80  6080      1T      0T 100% TRK       2 NO    1
     000B000B 000B000B   1 *-----1 WTESSNT.PRD.R921.EXP.CNT_____
     TRK       1 PO FB      80  3040      1T      0T 100% TRK       5 NO    1
     000C0002 000C0005   5 *-----4 UCIMSANT.GL9002.DATA_____
     TRK       4 PO FB      80  3120    170T      1T  99% TRK       4 NO    8
     000C0006 000C0007   1 *-----2 WTESSNT.PRD.R921.IDC.CNT_____
     TRK       2 PO FB      80  3040      2T      1T  50% TRK       2 NO    1
     000D0006 000D0006   1 *-----1 WTESSNT.OPS.ITS941.UTILCRD_____
     TRK       1 PO FB      80  6080      1T      0T 100% TRK      14 NO    1
     000D000B 000D000C   1 *-----2 UCIO010.ISPF350.ISPPROF_____
     TRK       2 PO FB      80  6160      2T      0T 100% TRK       1 NO    1
     000E0002 000E0005   4 *-----4 UCIMSANT.GL9002.DATA_____
     TRK       4 PO FB      80  3120    170T      1T  99% TRK       4 NO    8
```

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------------- VMAP STG00C   ----------- ROW 1 TO 17 OF 592
COMMAND ===>                                              SCROLL ===> CSR
Enter an ISPF command or a special control code:
  ---------------------------------------------------------------------------
CMD  CCHH    End CCHH EX# DATA/MSG ---------DATA SET NAME ------------------
     03E50000 05D0000E  1 *--7380 UCIMSAVT.GAP.APFM105.DATA_____
     05D10000 0716000A  1 *--4886 UCIHHCNT.HRCE0D.CUSENTDB_____
     088E0000 0968000E  1 *--3285 ACWCS50.OBB.T.EXDETAIL_____
     09EC0001 0A560001  1 *--1591 WTESSNT.PRD.Q941.APP.LOAD_____
     0C9F0000 0D02000E  1 *--1500 WFDFA1NT.TSO.JCL_____
     02510000 02AE0000  1 *--1396 WTESSNT.OPS.MIG.ISPFLIB_____
     0C270002 0C6E0000  3 *--1064 WFDFA1NT.PDB.STAR.NOV94_____
     0AB40000 0AEC000E  1 *---855 UCIGORNR.OREX0S.ORSEMIMO.STEP0010_____
     0A600000 0A960003  2 *---814 WFDFA1NT.PDB.STAR.NOV94_____
     01A70000 01DA000E  3 *---780 WTESSNT.OPS.MIG.ISPFLIB_____
     02E7000A 031B0003  1 *---774 WFDFA8NT.MM94206.MF9306.MF9304.BA9410_____
     0344000A 0375000E  1 *---740 *** FREE SPACE EXTENT ***_____
     09790000 09A80007  1 *---713 WTESSNT.PRD.R921.APP.LINK_____
     0C6E0001 0C9B000E  1 *---689 WFDFA7NT.CMIS94.HIST9410.CYCLEEND.PRODROLL_
     01770000 01A3000E  2 *---675 WTESSNT.OPS.MIG.ISPFLIB_____
     0AED0000 0B0E000E  1 *---510 WFDFA8NT.MM94206.MF9409.MF9404.BA9410_____
     02070000 02260006  1 *---472 *** FREE SPACE EXTENT ***_____
```

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
-----------------------------------------------------------------------------
PDSPN37 ----------------------- VMAP STG00C   ----------- ROW 1 TO 17 OF 583
COMMAND ===>                                              SCROLL ===> CSR
Enter an ISPF command or a special control code:
  ---------------------------------------------------------------------------
CMD  CCHH    End CCHH EX# DATA/MSG ---------DATA SET NAME ------------------
     03E50000 05D0000E  1 *--7380 UCIMSAVT.GAP.APFM105.DATA_____
     075C0000 0774000E  2 *---375 UCIMSAVT.GAP.APFM105.DATA_____
     07750000 078D000E  3 *---375 UCIMSAVT.GAP.APFM105.DATA_____
     078E0000 07A6000E  4 *---375 UCIMSAVT.GAP.APFM105.DATA_____
     07A70000 07BF000E  5 *---375 UCIMSAVT.GAP.APFM105.DATA_____
     05D10000 0716000A  1 *--4886 UCIHHCNT.HRCE0D.CUSENTDB_____
     088E0000 0968000E  1 *--3285 ACWCS50.OBB.T.EXDETAIL_____
     0BE10000 0BF8000E  2 *---360 ACWCS50.OBB.T.EXDETAIL_____
     02510000 02AE0000  1 *--1396 WTESSNT.OPS.MIG.ISPFLIB_____
     01770000 01A3000E  2 *---675 WTESSNT.OPS.MIG.ISPFLIB_____
     01A70000 01DA000E  3 *---780 WTESSNT.OPS.MIG.ISPFLIB_____
     010E0008 011A000C  4 *---185 WTESSNT.OPS.MIG.ISPFLIB_____
     00A90003 00C7000C  1 *---460 WFDFA1NT.PDB.STAR.NOV94_____
     0A600000 0A960003  2 *---814 WFDFA1NT.PDB.STAR.NOV94_____
     0C270002 0C6E0000  3 *--1064 WFDFA1NT.PDB.STAR.NOV94_____
     09EC0001 0A560001  1 *--1591 WTESSNT.PRD.Q941.APP.LOAD_____
     0C9F0000 0D02000E  1 *--1500 WFDFA1NT.TSO.JCL_____
```

VPRINT Subcommand

The VPRINT (VTAM print) subcommand prints a hardcopy list of a member. Use the TSO VPSPRINT or DSPRINT command (as generated during StarTool FDM installation). Add any VPSPRINT/DSPRINT operand after the member name.

Example

```
VPRINT   mema:memb  r525
```

Syntax

```
VPRINT   memgroup printer
         [CLASS(c)                                    ]
         [COL(col1:col2, ...)                         ]
         [COPIES(number)                              ]
         [DDNAME(ddname)                              ]
         [DIRECT(KEEP|DELETE)                         ]
         [EJECT|NOEJECT                               ]
         [FCB(fcbname)                                ]
         [FOLD(width)|TRUNCATE(width)                 ]
         [FORMS(formname)                             ]
         [HOLD|NOHOLD                                 ]
         [LINES(linenum1:linenum2)                    ]
         [NOEJMESS                                    ]
         [NOHEADER                                    ]
         [NUM(loc,length)|SNUM(loc,length)|NONUM      ]
         [PAGELEN(num)                                ]
         [SINGLE|DOUBLE|CCHAR                         ]
         [TERMINAL                                    ]
         [TMARGIN(num)|BMARGIN(num)                   ]
         [TRACE                                       ]
         [WTR(wtrname)                                ]

         Note: the above operands are for the VPSPRINT command.
```

Aliases

```
VP, VPR, VPRI, VPRIN, VPRINT
```

Defaults

CLASS(A), COPIES(1), NOEJECT, NOHOLD, NUM, SINGLE

Required

memgroup, printer

Operands

memgroup	Identifies the members to be printed.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names".
printer	Identifies the VTAM printer to print this data set.
operands	Optional. Can include any TSO VPSPRINT or DSPRINT operand.

Remarks

The VPRINT (VTAM print) subcommand prints a hardcopy list of a member.

Use the TSO VPSPRINT or DSPRINT command. This choice was made during StarTool FDM installation. To determine which program is used, type a CONTROL DEFAULT subcommand and look for an output line beginning "VPRINT calls".

VSAM Subcommand

The VSAM subcommand invokes the VSAM Services panel.

Syntax

VSAM

```
-------------------------------- VSAM Services -----------------------
OPTION ===>

   VINFO  - Information and Analysis, select a VSAM data set:
   Data Set name  ===>
                 VREPRO - REPRO using the above VSAM data set as input
                 VALLOC - Allocation of a VSAM data set
                 VMODEL - Allocation of a VSAM data set from model below,
                         If there is no model below, the VSAM data set above
                         will be used to initialize the information using a VINFO

Last information and analysis was for data set:
DSNAME:
TYPE:

Note:  All input fields are cursor selectable for the given service.
          Selecting the Data Set name will execute the VINFO service.

          "I" option from 3.2 will also initialize the model as will
          an INFO line command on a VSAM cluster in LISTC/LISTF
```

VTOC Subcommand

The VTOC subcommand displays selected data sets on a disk or set of disks. Each disk has a volume table of contents (VTOC). This table is searched for data sets that meet the specifications.

This interface is optional. Use it only if your installation installed the public domain VTOC command.

Example

VTOC tso001 CONT(dsna)

Syntax

```
VTOC   (volumelist)  [LEVEL(dsnamestart)              ]
                     [EXLEVEL(dsnamestart)            ]
                     [CONTAINING(dsnamestring)        ]
                     [ENDING(dsnameend)               ]
                     [BREAK(breakchars)               ]
              [CAT                                    ]
              [TRK|CYL|KB|MB                          ]
              [NOSORT|SORT(sortfields)                ]
              [NOPRINT|PRINT(printop (printitemlist)) ]
              [LIMIT(keyword oper value)              ]
              [AND1|OR1(keyword oper value)           ]
              [AND2|OR2(keyword oper value)           ]
              [AND3|OR3(keyword oper value)           ]
              [CHARS(charsperline)                    ]
              [LINES(linesperpage)                    ]
              [NOHEADING|HEADING(text)                ]
              [DSNLEN(length)                         ]
```

Aliases

VT, VTO, VTOC

Defaults

List all data sets on the volumes selected; SORT, PRINT, TRK

Required

volumelist

Operands

volumelist	A volume name or a list of volume names in parentheses. If you enter the first 1 to 5 characters of a volume name, all volumes that are mounted that start with those characters are listed. If you specify ALL, all non-virtual volumes that are online and ready are processed. If you specify ALV, all virtual volumes that are online and ready are processed if they are mounted PRIVATE/ RESERVED or PRIVATE/RESIDENT.
LEVEL (dsnstart)	The high-level qualifiers to be searched. This is not prefixed by your user ID or prefix. Only data sets starting with these prefixes are listed.
EXLEVEL (dsnstart)	The high-level qualifiers to be omitted. This is not prefixed by your user ID or prefix. Data sets starting with this prefix are not listed.
CONTAINING (dsnstr)	A character string contained in the data set name. At least one of the strings must be in the DSNAME for the data set to be listed. This string does not need to conform to DSNAME standards; it can begin with a period or a number.

ENDING (dsnameend)	The ending characters of the DSNAME. The final non-blank characters of the DSNAME must be one of these strings to allow the data set to be listed. This string must conform to DSNAME standards.
CAT	A locate is performed for each DSNAME on the volumes listed and status is indicated. This option can take a considerable amount of processing time. ■ C - cataloged on this volume ■ N - not cataloged ■ W - cataloged on another volume ■ E - catalog processing error
TRK	Space is to be reported in track units.
CYL	Space is to be reported in cylinder units.
KB	Space is to be reported in kilobyte units.
MB	Space is to be reported in megabyte units.
NOSORT	The data sets are not sorted. They are output as they are found.
SORT (sortfields)	Data sets are sorted into alphabetical order, based upon the sort fields specified. DSNAME, VOLUME, ALLOC, USED, UNUSED, PCT, EX, DSO, RFM, LRECL, BLKSZ, CDATE, EXPDT, and REFDT are valid sort fields. An A or D is required after each sort field to indicate ascending/descending sequence.
BREAK (breakchars)	The listing contains a new header (on a new page if you use the VTOCOUT DD card option) whenever the specified number of characters differs from the preceding data set. This option functions only with the SORT option.
CHARS (charsperline)	The number of characters on each line of output. The default is 150 for print and the line width of the terminal for TSO sessions. You can get more information by specifying a larger number of characters per line or you can limit the printing by setting a smaller number of characters per line of output.
LINES (linesperpage)	Specifies the number of lines before a new title line is produced. It defaults to 60 for print and to the screen size for TSO sessions.
NOHEADING	Does not produce a heading. The heading is only output if the VTOCOUT DD statement is present.
HEADING(text)	If a DD statement with a DDNAME of VTOCOUT is present, this text is used to begin every page. Include ASA carriage control. The default header consists of "VTOC command version 02" followed by the command that was entered.
DSNLEN (length)	Specifies the length of the DSNAME to print. The rest of the DSNAME is truncated. The CHARS parameter also causes the DSNAME to be truncated if the name and the preceding information exceeds the print line.
NOPRINT	Specifies that individual items are not to be listed. Use the command to calculate totals.

PRINT(ptop (ptitems))	The items to print. Ptop is the print formatting option as follows: ■ **NEW -** the ptitems is a complete list of what to print ■ **REP -** the first ptitem is replaced with the rest of the items on the list ■ **ADD -** the rest of the ptitems are added after the first item on the list ■ **DEL - t**he ptitems named are not printed The ptitems names are the same keywords used in LIMIT, AND, and OR functions and are also the titles as printed. These keywords can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. The ADD, DEL, and REP print operations refer to the default print list. The default list is ALLOC, UNUSED, PCT, EX, DSO, RFM, BLKSZ, LRECL, REFDT, CDATE, VOLUME, DSNAME, EXPDT, SECQ, SECT, ROUND, PASS, ACTION, and type. As noted under CHARS above, only the items that fit on the print line are listed.
LIMIT(key oper value)	The data sets to be listed. Only data sets that satisfy the relation are listed. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.

keyword	The name of a data set field as follows. The keywords and their values are the same as in the VTOC output.
	■ ALLOC - data set allocation; number of kilobytes, tracks, cylinders, or megabytes allocated. Default is in tracks
	■ UNUSED - amount of unused space in the data set. Same units as in ALLOC
	■ USED - amount of space used in the data set. Same units as in ALLOC
	■ PCT - percentage of space used in the data used
	■ EX - number of extents in the data set
	■ DSO - data set organization PS = sequential PO = partitioned VS = VSAM PE = PDSE data DA = direct IS = ISAM U = unmoveable
	■ **RFM** - record format F = fixed V = variable U = undefined B = blocked T = track overflow S = spanned or standard A = ASA carriage control M = machine carriage control
	■ BLKSZ - block size for physical blocks of data
	■ LRECL - logical record length in bytes
	■ CDATE - creation date in the form YYDDD, sometimes called Julian

	■ EXPDT - expiration date in the same form. This field is rarely used
	■ REFDT - last use date in the same form. This date is when the data set was last opened
	■ SECT - type of allocation A = absolute track B = blocks T = tracks C = cylinders
	■ PASS - protection indicators N = none T = read and write protection W = write protection
	■ ROUND - space rounded up to cylinders R = round N = no round
	■ CCHH - cylinder and head address, in 4 or 8 hexadecimal digits. If 4 digits are used, only the cylinder is used for comparison; otherwise, the cylinder and track are compared
	■ VOLUME - Volume serial number or disk name
	■ UNIT - Unit or device type
	■ DSNAME - Name of the data set
	■ ACTION - Some error indications
	■ TYPE - Reserved for exit usage
	■ RACF - RACF indicator N = not indicated Y = indicated
	■ UPD - updated since last backup N = not updated Y = updated
oper	An operator. The list of operators is as follows: ■ EQ is equal to ■ NE is not equal to ■ LE is less than or equal to ■ LT is less than ■ GE is greater than or equal to ■ GT is greater than
value	Gives the value of the item for comparison, such as FB, PS, R, or a number.
AND1(key oper value)	Data sets to be listed. Both LIMIT and this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.

OR1(key oper value)	Data sets to be listed. Either LIMIT and this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.
AND2(key oper value)	Data sets to be listed. Both the previous result and this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.
OR2(key oper value)	Data sets to be listed. Either the previous result or this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.
AND3(key oper value)	Data sets to be listed. Both the previous result and this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.
OR3(key oper value)	Data sets to be listed. Either the previous result or this condition must be true to allow the listing. *key* can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF, or UPD. *oper* can be EQ, NE, LE, LT, GE, or GT. *value* can be a comparison value such as FB, PS, R, or a number like 51.

Remarks

Following are some examples of uses for the VTOC command.

- List all data sets on VOL*** volumes:

    VTOC vol

- List all data sets on all volumes:

    VTOC all

- List all data sets that start with XXX on any MV**** volume:

VTOC mv lev(xxx)

- List all data sets that contain LIST and do not start with SYS1:

    VTOC vol cont(list) exlev(sys1)

- List all recently created data sets:

    VTOC vol lim(cdate gt 93001)

- List data sets with unused space:

    VTOC vol lim(pct lt 50) or1(unused gt 30)

- List data sets with multiple extents:

    VTOC vol lim(ex gt 1)

- List all of a user's data sets on TSO001:

    VTOC tso001 level(ser002)

- List all data sets over 100 tracks:

    VTOC vol lim(alloc gt 100)

- List CLIST data sets:

    VTOC vol end(clist)

- List all of the information about a user's data sets at a 3270:

    VTOC vol char(150)

- List data sets under the fixed heads:

    VTOC vol lim(cc eq 0001) or1(cc eq 0002)

- Check if data sets on volume VOL*** are cataloged correctly:

    VTOC vol cat print(new (alloc pct cat dsname))

- List used space instead of unused, in tracks:

    VTOC vol print(rep (unused used)) trk

- List all data sets on 335*** volumes sorted by ALLOC in descending sequence, VOLUME and DSNAME in ascending sequence:

    VTOC 335 sort(alloc,d,volume,a,dsname,a)

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------ ISPMODE Session# 1 Log# 1 --  ROW 112 TO 128 OF 128
COMMAND ===>                                            SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY -------------------------
>----->vtoc ser cont(ser07.lib.)
  ALLOC  UNUSED PCT EX DSO RFM LRECL BLKSZ CDATE REFDT VOLUME DSNAME
    379        100  2 PO  FB    80 13680 93064 93197 SER006 SER07.LIB.ASM
    140     97  30  4 PO  FB    80  9040 89277 93207 SER007 SER07.LIB.CLIST
      4      2  50  3 PO  VB   255 32760 91074 93206 SER006 SER07.LIB.CLISTV
    100     15  85  1 PO  FB    80 13680 93076 93207 SER006 SER07.LIB.CNTL
      1        100  1 PO  FB    80  9040 92199 93015 SER006 SER07.LIB.EXPDT
     25        100  1 PO  FB    80  9040 89277 93194 SER007 SER07.LIB.FILE11
     63        100  1 PO  FB    80  9040 89277 93204 SER007 SER07.LIB.FILE29
    203        100  2 PO  U      0 32000 89277 93195 SER007 SER07.LIB.LOAD
    765    225  70  3 PO  FB    80 13680 93126 93207 SER006 SER07.LIB.PDSE
    459        100  1 PO  FB    80 13680 93126 93204 SER006 SER07.LIB.PDSE31
    180        100  2 PO  FB    80 13680 93180 93181 SER006 SER07.LIB.PDSE32
      4        100  1 PO  FB    80  9040 92265 93176 SER007 SER07.LIB.PDSINS
      2        100  1 PO  U      0 13030 93190 93197 SER007 SER07.LIB.STOW
     10      9  10  1 PS  FB   200  2000 93181 93181 SER007 SER07.LIB.TESTVS
 TOTALS -   14 DATA SETS,    2335 TRKS   ALLOC,     1987 TRKS   USED
****************************** BOTTOM OF DATA ******************************
```

VUSE Subcommand

The VUSE (volume usage) subcommand checks disk volume statistics and usage.

Example
```
VUSE   sysres  all
```

Syntax
```
VUSE   volume [SHORT|LONG|ALL]
```

Aliases
```
VU, VUS, VUSE
```

Defaults

SHORT

Required

volume

Operands

SHORT	Provides basic information for a volume including free space statistics and the current volume usage.
ALL	Includes SHORT information and formats any SMS VOLUME RECORD information and the FORMAT 4 DSCB. Same as LONG.
LONG	Includes SHORT information and formats any SMS VOLUME RECORD information and the FORMAT 4 DSCB. Same as ALL.

Remarks

The VUSE subcommand is useful for determining the amount of free space on a volume as it displays the largest five free extents in track and cylinder units. It is also useful for diagnosing volume errors as the FORMAT 4 DSCB can be interpreted by this subcommand.

```
------------------------- ISPMODE Session# 1 Log  ROW 1,223 TO 1,234 OF 1,234
COMMAND ===>                                          SCROLL ===> CSR
- DSN=WSER07.SMSLIB.PDSETEST,VOL=SER=STG005  MEM=$$$$CUST -------------------
>----->vuse stg00c
PDS082I Volume name: STG00C     UNIT = 5C4     TYPE = 3390M3
PDS083I Volume status: RESIDENT   PRIVATE  ALLOCATED    ONLINE
PDS085I Blank DSCB's: 4927 or 86%
PDS086I Free indexed VTOC VIR's: 577

PDS087I Free space:  5560 TRACKS OR 23%; 112 EXTENTS INCLUDING  342 FULL CYLIN

PDS089I LARGEST EXTENTS:      #1       #2       #3       #4       #5
PDS089I         CYL.TRKS    49.05    31.07    30.12    25.11    17.08
PDS089I         TRACKS       740      472      462      386      263

PDS096I DEVICE  MB/VOL   TRACKS   #CYLS  TRK/CYL  BYTES/TRK  DSCB/TRK  PDS/TRK
PDS096I 3390M3   2,838   50,085   3,339      15     56,664        50       45
```

```
------------------------- ISPMODE Session# 1 Log  ROW 1,121 TO 1,139 OF 1,222
COMMAND ===>                                           SCROLL ===> CSR
- DSN=WSER07.SMSLIB.PDSETEST,VOL=SER=STG005  MEM=$$$$CUST -------------------
>----->vuse stg00c all
PDS082I Volume name: STG00C     UNIT = 5C4     TYPE = 3390M3
PDS083I Volume status: RESIDENT   PRIVATE  ALLOCATED    ONLINE
PDS085I Blank DSCB's: 4927 or 86%
PDS086I Free indexed VTOC VIR's: 577

PDS087I Free space:  5560 TRACKS or 23%; 112 EXTENTS INCLUDING  342 FULL CYLIN
PDS089I LARGEST EXTENTS:       #1      #2      #3       #4       #5
PDS089I        CYL.TRKS    49.05   31.07   30.12    25.11    17.08
PDS089I         TRACKS      740     472     462      386      263

PDS096I DEVICE  MB/VOL   TRACKS   #CYLS  TRK/CYL  BYTES/TRK  DSCB/TRK  PDS/TRK
PDS096I 3390M3   2,838   50,085   3,339      15     56,664        50       45

PDS185I Format 4 DSCB at 0002000001
PDS185I 0   4   8   C   10  14  18  1C  20  24  28  2C  30  34  38  3C  40  44
PDS185I ...........................................4......".......a......Vs..
PDS185I 000000000000000000000000000000000000000000000F000031700000080C00000EA00
PDS185I 444444444444444444444444444444444444444444444080E24FDB000F1100DC0F5200

PDS185I  48  4C  50  54  58  5C  60  64  68  6C  70  74  78  7C  80  84  88
PDS185I ...................................................................
PDS185I 020032000000000000000000000000000000000000000000000000000000000000000000
PDS185I 00002D000000000000000000000000000000000100200080E00000000000000000000000

PDS186I LOC NAME     VALUE         DESCRIPTION
PDS186I --- ----     -----         -----------
PDS186I 00  DS4KEY   04040404...   KEY - VTOC NAME - 44X'04'
PDS186I 2C  DS4IDFMT F4            FORMAT IDENTIFIER
PDS186I     DS4HPCHR 0008000E32    HIGHEST FORMAT 1 DSCB (NOT MAINTAINED)
PDS186I     DS4DSREC 5,247.        NUMBER OF AVAILABLE DSCB'S (NOT MAINTAINED)
PDS186I 34  DS4HCCHH 0D0B0000      CCHH OF NEXT AVAILABLE ALTERNAME TRACK
PDS186I 38  DS4NOATK 15.           NUMBER OF REMAINING ALTERNATE TRACKS
PDS186I 3A  DS4VTOCI 81            VTOC INDICATORS
PDS186I                   X'81' DS4IVTOC - THIS VOLUME HAS AN INDEXED VTOC
PDS186I 3B  DS4NOEXT 1.            NUMBER OF EXTENTS IN THE VTOC
PDS186I 3C  DS4SMSFG 0000          SYSTEM MANAGED STORAGE INDICATORS
PDS186I 3E  DS4DSCYL 3,340.        NUMBER OF LOGICAL CYINDERS
PDS186I 40  DS4DSTRK 15.           NUMBER OF TRACKS IN A LOGICAL CYLINDER
PDS186I 42  DS4DEVTK 58,786.       DEVICE TRACK LENGTH
   . . .
PDS186I 4A  DS4DEVDT 50.           NUMBER OF DSCB'S PER TRACK
PDS186I 4B  DS4DEVDB 45.           NUMBER OF DIRECTORY BLOCKS PER TRACK
PDS186I 4C  DS4AMTIM 0000000000000000  VSAM TIME STAMP
PDS186I 54  DS4VSIND 00            VSAM INDICATORS
PDS186I 55  DS4VSCRA 0000          RELATIVE TRACK LOCATION OF THE CRA
PDS186I 57  DS4R2TIM 0000000000000000  VSAM VOLUME/CATALOG MATCH TIME STAMP
PDS186I 5F          0000000000     RESERVED
PDS186I 64  DS4F6PTR 0000000000    CCHHR POINTER TO FIRST FORMAT 6 DSCB OR ZERO
PDS186I 69  DS4VTOCE 0100000200    CCHHR ADDRESS OF VTOC EXTENT START
PDS186I 6E  DS4VTOCE 000008000E    CCHHR ADDRESS OF VTOC EXTENT END
```

If a volume is SMS-controlled, the ALL or LONG operand also formats the following SMS VOLUME RECORD information.

```
PDS088I Volume record definition dump:
        0000  C9C7C4E5 D3C44040  00000000 00000001  *IGDVLD  ........*
        0010  00094040 00000098  0006E2E3 C7F0F0C3  *..  ...q..STG00C*
        0020  00000000 00000000  00000000 00000000  *................*
        0030  00000000 00000000  C1C3E2E2 C9F0F940  *........ACSSI09 *
        0040  F1F9F9F4 61F0F661  F1F7DD58 00000000  *1994/06/17......*
        0050  F1F57AF3 F3404040  0004E2E3 D9C70000  *15:33   ..STRG..*
        0060  00000000 00000000  00000000 00000000  *................*
        0070  00000000 00000000  01010102 01010101  *................*
        0080  00000000 00000000  00FADC08 00000A93  *...............l*
        0090  0000012C 00000027  0000008B 091305E8  *...............Y*
        00A0  01114928 0000C1E7  01040104 00000000  *......AX........*

PDS186I 38  VLDDUSER ACSSI09       USERID OF LAST UPDATER
PDS186I 40  VLDDDATE 1994/06/17    DATE OF LAST UPDATE
PDS186I 4A  VLDTRKSZ 56,664.       VOLUME R1 TRACK CAPACITY
PDS186I 50  VLDDTIME 15:33         TIME OF LAST UPDATE
PDS186I 5A  VLDSTGRP STRG                         VOLUME STORAGE GROUP NAME
PDS186I 78  VLDSMSS  01            (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 79  VLDMVSS  01            (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 7A  VLDSMSS  01            (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7B  VLDMVSS  02            (VLDOFFLN) MVS STATUS IS OFFLINE
PDS186I 7C  VLDSMSS  01            (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7D  VLDMVSS  01            (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 7E  VLDSMSS  01            (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7F  VLDMVSS  01            (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 88  VLDNUCBA 00FADC08      ADDRESS OF UCB IF KNOWN OR ZERO OTHERWISE
PDS186I 8C  VLDNTCPY 2,707.        TOTAL CAPACITY IN MEGABYTES
PDS186I 90  VLDNFREE 300.          AMOUNT FREE IN MEGABYTES
PDS186I 94  VLDNLEXT 39.           LARGEST FREE EXTENT IN MEGABYTES
PDS186I 98  VLDFLAGS 00            SMS CONVERSION FLAGS
PDS186I 9A  VLDN0CNT 139.          VOLUME LEVEL RESET COUNT
PDS186I A0  VLDSGST  01            STORGRP STATUS ON THIS SYSTEM
PDS186I A4  VLDNLEVL 49,639.       UPDATE LEVEL FOR VOLUME
PDS186I A8  VLDCSMSS 0104010400000000   CONFIRMED SMS STATUS FOR VOLUME
```

WHOHAS Subcommand

The WHOHAS subcommand displays users allocated to a data set.

Example
```
WHOHAS   'sys1.uads'
```

Syntax
```
WHOHAS   dataset | *
```

Aliases
```
WH, WHO, WHOH, WHOHA, WHOHAS
```

Defaults

Required

dataset

Operands

dataset	Identifies the data set to be checked.
*	Check the currently allocated data set.

Remarks

Use WHO (short for WHOHAS) as a line command in LISTA/DDNAME, LISTC/LISTF and WORKPAD.

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
------------------------ ISPMODE Session# 1 Log  ROW 1,000 TO 1,005 OF 1,005
COMMAND ===>                                               SCROLL ===> CSR
- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011  ----------------------
>----->whohas 'sys1.uads'
PDS292I 'SYS1.UADS' is allocated as follows:
PDS292I JOBNAME  SCOPE   TYPE   STATUS  SYSTEM  RESERVE
PDS292I DLE1     SYSTEM  SHR    USING   SCU1    NO
PDS292I SSI003   SYSTEM  SHR    USING   SCU2    NO
PDS292I SSI002   SYSTEM  SHR    USING   SCU1    YES
PDS292I SER07    SYSTEM  SHR    USING   SCU2    CONVERTED
****************************** BOTTOM OF DATA ******************************
```

WORKPAD Function

Use the WORKPAD command to store StarTool FDM subcommands, data set names, TSO commands, CLISTS, and REXX execs in an ISPF table. Each individual table element can be re-executed whenever you want.

WORKPAD tables are saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE and ERASE commands. To automatically save these tables at program end or when a new WORKPAD table is retrieved, check the setting of "Automatic save" in SETALL for WORKPAD.

Saved WORKPAD tables are given members names of the form PDSWP0na if the name is one or two numeric digits; otherwise, use a name of the form @@name where name is a 1- to 6-character alphanumeric table identification name.

Example

WORKPAD

Syntax

WORKPAD   [name]   [ALTERNAT]   [SETUP]

Aliases

W, WO, WOR, WORK, WORKP, WORKPA, WORKPAD

Defaults

0

Required

Operands

name	1- to 6-alphanumeric characters, the WORKPAD table name to retrieve.
ALTERNAT	Displays the alternate WORKPAD panel (for new WORKPADs only).
SETUP	Execute each line of the WORKPAD in EXPRESS mode.

Remarks

The WORKPAD ISPF table displays in response to a WORKPAD command. When you are in a WORKPAD display, you can delete a part of the table, find data in the table, insert elements into the table, print a part of the table, store a part of the table in a data set, and so on. For help with the different options available, use the HELP command, CUA functions, or enter the O command as a primary command or as a line command.

WORKPAD is a versatile service to keep sets of commands and data set names for easy reference. These sets are shared between groups and the installation. Sample table 99 is supplied by Micro Focus and can be modified by your installation as an index to other samples. Reserve table 0 for temporary use and table 1 for your own index. Use Table 2 for setup.

The SETUP operand on the WORKPAD command allows you to execute a series of StarTool FDM commands without intervention. Any data set entries in this mode result in executing a GO line command. Follow each data set entry by a MEMLIST command to build a MEMLIST for each data set.

When you modify a WORKPAD table entry, it is not executed automatically. You can change this default for the current session by typing MODE EXEC.

You can also change this default across sessions in SETALL.

Each WORKPAD element or line has the following four fields:

OPT	Option. Specifies the program action (the line command field).
MSG/RC	Message or return code. Displays program feedback messages.
TYPE	Command type. Specifies the type of entry. ■ * is a comment entry ■ - is a TSO command, a CLIST, or a REXX exec ■ D is a data set name ■ P is a StarTool FDM subcommand ■ T is a TSO command whose output is to be TRAPPED in the log
PDS/TSO	Command field. This is where the data set name, command, or subcommand is entered and displayed. Comments can be added after any entry. Type **/*** and follow it with any data.

The alternate WORKPAD table contains additional entry fields for data sets. Enter the volume name and a member group specification with up to 17 characters. The BB block line command and the BR, ED, and ML line commands use the MEMBERS field from the alternate WORKPAD table.

The following primary commands are supported for the WORKPAD function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" on page 37.

ADD	Adds 5 blank table lines to the end of the WORKPAD table.	
ALT[ERNAT]	Displays an alternate view of the WORKPAD table.	
APP[LY]	Applies the specified line command to all table entries and executes each entry.	
	Syntax	APPLY linecmd
COMPR[ESS]	Global command. Changes to each data set in the table and issues a COMPRESS subcommand.	
	Syntax	COMPRESS [LIST|NOLIST|SUMMARY]
DUA[L]	Displays a double line view of the WORKPAD table.	
EDITT[BL] (or ET[BL])	Enters an edit session on WORKPAD table data.	
ER[ASE]	Deletes the WORKPAD table in memory and on disk.	
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.	
F	Finds a string and positions the display start location.	
	Syntax ```F anystring [ASIS] [FIRST|LAST|PREV] [PREFIX|SUFFIX|WORD] [ANY|COMMAND]```	
FI[ND]	Global command. Changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is applied to each data set individually, do not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.	

	Syntax
	`FIND memgroup 'anystring'`
	`        [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]`
	`        [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]`	

GL[OBAL]	Global command. Changes to each data set in the table and issues any StarTool FDM subcommand.	
	Syntax	`GLOBAL anysubcommand [anyoperands]`
ID	Changes the table ID name.	
	Syntax	`ID name`
	(where name is a 1- to 6-character alphanumeric name).	
INS[ERT]	Adds blank WORKPAD table lines.	
	Syntax	`INSERT [nn] [TOP|BOTTOM|`<u>`HERE`</u>`]`
	(where *nn* defaults to 5)	
LE[FT]	Rotates through alternate views of the WORKPAD table. PF keys 10 and 22 are normally set to LEFT.	
LO[AD]	Loads data set names from operating system control blocks.	
	Syntax	`LOAD APFLST|LNKLST|LPALST [RESET]`
MER[GE]	Merges data from a different table into the current table.	
	Syntax	`MERGE name [GROUP|SAMPLE] [RESET]` `        [TOP|BOTTOM|HERE]`
	(where name is 1- to 6-alphanumeric characters)	
MODE	Specifies if modified WORKPAD lines are to be automatically executed.	
	Syntax	`MODE [EXEC|NOEXEC]`
MODEL	Global command. Changes to each data set in the table and issues a MODEL command.	
NOR[MAL]	Displays the default view of the WORKPAD table.	
NOS[AVE]	Do not save the current WORKPAD on disk regardless of the setting of "Automatic save" in SETALL for WORKPAD.	
O[PTIONS]	Provides primary command selection for the WORKPAD function and operand syntax assistance.	
OUT[PUT]	Outputs the WORKPAD table to print or a data set.	
	Syntax	`OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the WORKPAD table based on a string match.	

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [ANY|COMMAND]`	

REPL[ACE]	Global command. Changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is applied to each data set individually, do not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.	

	Syntax
	```\nREPLACE memgroup 'fromstring' 'tostring'\n [NUM|SNUM|NONUM|LBLOCK|LDUMP|BLOCK|DUMP]\n [CAPS|ASIS|IGNORE] [WORD|PREFIX|SUFFIX]\n [WRITE|NOWRITE]\n```
RESET	Often used with data set tagging, clears the DATA/MSG field in all table entries.
RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.
RI[GHT]	Rotates through alternate views of the WORKPAD table. PF keys 11 and 23 are normally set to RIGHT.
SAM[PLE]	Merges table number 99 from the installation table library (usually ISPTLIB) into the current WORKPAD table. Table 99 is an index to other sample WORKPAD tables. The additional sample tables can be included by using the MERGE command.
SAVE	Creates a permanent table for use in a later StarTool FDM session.
	Syntax `SAVE [name] [REPLACE|NOREPL]`
	(where *name* is 1- to 6-alphanumeric characters)
SEEK	Global command. Changes to each data set in the table and looks for a member.
	Syntax `SEEK member`
SO[RT]	Sorts the WORKPAD table into order based on the WORKPAD entries.
	Syntax `SORT`
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
	Syntax `TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation.
WHO[HAS]	Global command. Issues a WHOHAS subcommand for each data set in the table.
X	Clears the WORKPAD table relative to the cursor position.
	Syntax `X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the WORKPAD table; this is equivalent to X ALL.

The following line commands are supported in the WORKPAD function. The DFHSM commands are supported from the UT panel and BR, ED and ML refer to a MEMBERS column displayed on an alternate panel that can be reached with PF10 or PF11.

+	Provides an extension panel for command entry. `*cm ERR- t="Line command,+">`
=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
6	Changes to the data set and issues a MEMLIST all.
A	After command. Copies or moves data after the marker.

ALT	Changes to the data set and executes an alternate subcommand.
B	Before command. Copies or moves data before the marker.
BR	Changes to the data set and issues a MEMLIST for members in the MEMBERS column.
C	Copies this line.
CH	Changes to the data set.
COMB	Changes to the data set and invokes COMBINE.
COPY	Changes to the data set and invokes COPY.
DUP	Changes to the data set and invokes DUP.
ED	Changes to the data set and issues a MEMLIST for members in the MEMBERS column.
EXEC	Executes the command, subcommand, CLIST, REXX exec, or changes to the data set.
FIND	Changes to the data set and invokes FIND.
FIX	Changes to the data set and invokes FIXPDS.
HBAC	HBBACK DFHSM command to back up the data set.
HBDE	HBDEL DFHSM command for the data set.
HDEL	HDEL DFHSM command for the data set.
HIST	Changes to the data set and invokes HISTORY.
HMIG	HMIG DFHSM command for the data set.
HML2	HMIG DFHSM command for the data set to LEVEL 2.
HREC	HRECALL DFHSM command for the data set.
GO	Changes to the data set using GO processing (a number is optional).
IF	Changes to the data set and invokes IF.
IN	Inserts a blank line.
K	Kills and clears all following line commands.
LC	Adds the data set name to the current LISTC/LISTF table.
LOG	Copies the line into the log.
M	Moves this line.
MAP	Changes to the data set and invokes MAP.
ME	Provides line command selection and entry assistance.
ML	Changes to the data set and issues a MEMLIST for members in the MEMBERS column.
MOD	Changes to the data set and issues a MODEL command.
O	Provides line command selection and operand syntax assistance.
PBRO	Changes to the data set and invokes PBROWSE.
PEDI	Changes to the data set and invokes PEDIT.
PVIE	Changes to the data set and invokes PVIEW.

R	Reproduces this line.
REPL	Changes to the data set and invokes REPLACE.
S	Selects line command (normally EXEC, it is set in SETSEL).
SEEK	Changes to the data set and checks for a member.
SEPA	Changes to the data set and invokes SEPARATE.
SETA	Saves a StarTool FDM subcommand for repeated use by the ALT line command.
SMPG	Changes to the data set and invokes SMPGEN.
SUBL	Changes to the data set and invokes SUBLIST.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
UP	Updates or views, then executes this entry.
US	Changes to the data set and displays USAGE.
UT	Selects the extended user line command panel.
VERI	Changes to the data set and invokes VERIFY.
VIEW	Views and updates but does not execute this entry.
WHO	Checks for users of this data set with WHOHAS.
X	Drops the table line.
Z	Changes to the data set and issues a COMPRESS subcommand.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the WORKPAD function:

==	= command. Repeats the previous line command for the range of lines. *cm ERR- t="Block line command,==">
AA	ALT command. Changes to each data set and executes an alternate subcommand.
BB	BR command. Changes to each data set and issues a MEMLIST for the MEMBERS specified.
CC	C command. Copies this range of table entries.
EE	EXEC command. Executes each entry.
GG	GO command. Changes to each data set using GO processing.
LL	LOG command. Copies the range of lines into the log.
MM	M command. Moves this range of table entries.
OO	O command. Provides line command assistance for each line.
RR	R command. Reproduces this range of table entries.
SS	S command. Selects each line in the range of table lines.
SSEE	SEEK command. Changes to each data set and check for a member.
UU	UP command. Updates or views then executes each entry.
VV	VIEW command. Views and changes each entry but does not execute.

XX	X command. Drops the range of table lines.
ZZ	Z command. Changes to each data set and issues COMPRESS subcommand.

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------- StarTool Workpad Table 0 --------- ROW 1 TO 4 OF 4
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER005  MEM=/ASM  -------------------------------
OPT  MSG/RC  TYPE ------ PDS/TSO COMMAND or CLIST --------------------------
---- -------- -  -------------------------------------------------------------
            *    This whole entry is a comment.

---- -------- -  -------------------------------------------------------------
            P    list pds99t1d       /* this is a comment

---- -------- -  -------------------------------------------------------------
     *RC=4   T    dsat lib.cntl       /* this is a comment

---- -------- -  -------------------------------------------------------------
            D    'sys1.parmlib'      /* this is a comment

---- -------- -  -------------------------------------------------------------
```

```
 FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------- StarTool Workpad Table 15 -------- ROW 1 TO 4 OF 4
COMMAND ===>                                              SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER005  MEM=/ASM  -------------------------------
OPT  MSG/RC  TYP MEMBER  VOLUME ------- DSNAME/PDS/TSO COMMAND or CLIST ------
            D                  'SER07.SMSTEST.DATA06'
 ut  *ML*    D @:ABC   MVSD4B 'SER07.SMSTEST.DATAPDS'
            D          MIGRAT 'SER07.PDS110.DOC'
     *EXEC*  P                list pds99t1e    /* this is a comment
```

```
-------------------- ut for WORKPAD   user line commands --------------------
OPTION  ===>
Choose one of the following for DSN type entries:
    HDEL - HDEL command for data set 'SER07.SMSTEST.DATAPDS'
    HMIG - HMIG command for data set 'SER07.SMSTEST.DATAPDS'

Dynamic Commands--note:  < = MVSD4B,
                         / = 'SER07.SMSTEST.DATAPDS'
LAST - TRP(DSAT / LAST) /* GET THE LAST REFERENCE DATE
LCLC - TRP(LISTC ENT(/) ALL)
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
```

XREF Subcommand

The XREF subcommand lists internal symbol cross references in a load module. It is similar to the output produced by the XREF option of the linkage editor except that references are provided by actual name used and sorted by name or by location.

Example

```
XREF    mema:memb
```

Syntax

```
XREF    memgroup [MODULE({* | Fullm | Partm*)} ]
                  [ENTRY(Entname)            ]
                  [SHORT                     ]
                  [SORT|NOSORT               ]
                  [NOSTACK                   ]
```

Aliases

```
XR, XRE, XREF
```

Defaults

memgroup, SORT

Required

Operands

memgroup	Identifies the load members for which you want cross reference information.
	Default member names, member lists, member name ranges, and member name patterns are allowed. For more information, see Appendix A, "Formatting Member Names" on page 421.
MODULE (name)	A1- to 8-byte partial external name that limits CSECT and ENTRY names for XREF reporting.
	The MODULE operand has several valid forms:
	■ MODULE(*) - uses the previous name entered on any MODULE keyword
	■ MODULE(Fullm) - reports only on CSECT or ENTRY FULLM
	■ MODULE(Partm*) - reports only on CSECT or ENTRY PARTM...
ENTRY (Entname)	A 1- to 8-byte partial entry name used to limit XREF output similar to the MODULE keyword.
SHORT	Displays only the callers of a CSECT or ENTRY point name.
SORT	Output CSECT names in order of CSECT name.
NOSORT	Output CSECT names in order of address or CSECT location.
NOSTACK	Outputs a single data value on an output line for use with programs that post-process XREF outputs.

Remarks

The XREF subcommand lists internal symbol cross references in a load module.

In the following example, the PDS441W message documents a missing weak external reference. The missing CSECT (VTSOCMD) is referenced from PDSMAIN and PDSPCOMM.

Also, on the PDS168I message, the CSECT name and any ENTRY name referenced is noted with a special syntax. In this example, PDSMAIN has at least one reference to ENTRY PDS$CHA in CSECT PDSOPTIO. This information is represented by PDSOPTIO<PDS$CHA> in the second line of the first PDS168I message.

```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------- ISPMODE Session Display ---------- ROW 563 OF 726
COMMAND ===>                                              SCROLL ===> CSR
- DSN=C911407.SAM.LOAD,VOL=SER=STR807  MEM=PDSE  ----------------------------
>----->xref pdse nosort

** XREF     PDSE
PDS441W PDS#SECI (Weak)
PDS441W VTSOCMD (Weak)
PDS166I PDS#SECI From: PDSMAIN
PDS166I VTSOCMD From: PDSMAIN, PDSPCOMM
PDS166I PDSMAIN From: PDSSTAEX, PDSAPPL, PDSOPTIO, PDSIDSPY, PDSPCOMM
PDS168I PDSMAIN To: PDS#SECI, VTSOCMD, PDSPDSIN, PDSATTNX, PDSSTAEX, PDSPARSE,
        PDSIDSPY, PDSALLOC, PDSEXCP, PDSCLEAR, PDSPCOMM, PDSOPTIO<PDS$CHA>,
        PDSDSNAM, PDSEXEC, PDSOPTIO<PDS$TBL>, PDSOPTIO, PDSDISPL,
        PDSLIST<PDSREPLA>, PDSLIST<PDSLISTX>, PDSMSGS
PDS166I PDSPDSIN From: PDSMAIN
PDS166I PDSATTNX From: PDSMAIN
PDS166I PDSSTAEX From: PDSMAIN, PDSPCOMM, PDSIDSPY
PDS168I PDSSTAEX To: PDSMAIN, PDSALIAS, PDSPARSE, PDSIDSPY
PDS166I PDSALLOC From: PDSMAIN, PDSPCOMM
```

```
------------------------- ISPMODE Session# 1 Log# 1 ----  ROW 45 TO 64 OF 368
COMMAND ===>                                              SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
- DSN=WSER07.LINK.LOAD,VOL=SER=STR98A  MEM=PDSE420  -------------------------
 >----->xref pdsexref
** XREF     PDSEXREF
 PDS540W CSECT ALIASES is not referenced
 PDS540W CSECT COMPARED is not referenced
 PDS166I PDS#DFLS From: PDSCONTR
 PDS166I PDS#OPT4 From: PDSPDSIN
 PDS166I PDS#SECI From: PDSCONTR, PDSPDSIN
 PDS166I PDSALIAS From: PDSOPTIO, PDSSTAEX
 PDS166I PDSALLOC From: PDSCOMPR, PDSFIXPD, PDSMAIN
 PDS166I PDSATTRI From: PDSOPTIO
 PDS166I PDSBROWS From: PDSOPTIO
 PDS166I PDSCALCT From: PDSIDSPY, PDSSPACE
 PDS166I PDSCHANG From: PDSFIXPD, PDSOPTIO, PDSPCOMM
 PDS166I PDSCLEAR From: PDSMAIN, PDSRESTO
```

ZAP Function

The ZAP command modifies load modules. Make these modifications by typing over hexadecimal or character data. ZAP is a line command supported by the CSECTS command.

Example

```
zap
```

Syntax

```
zap
```

Aliases

```
z, za, zap
```

Defaults

Required

Operands

Remarks

The CSECTS ISPF table displays in response to a CSECTS command. The ZAP line command invokes the ZAP command processor for a given CSECT in a load module. When you are in a ZAP display you can delete a part of the table, find data in the table, print a part of the table, or store a part of the table in a data set. For help with the different options available, use the HELP command, CUA functions, or enter an O command as a primary command or as a line command.

To make modifications to the ZAP table, type over the character or hexadecimal data. After you make all changes to a CSECT, enter ZAP as a primary command to update your on disk. Each ZAP data line has the following fields:

CMD	Command. This is where line commands are entered.
MODULE-ADDR	Module address. This is the offset in the current module.
BASE-OFFSET	Base offset. This is the offset in the current CSECT.
MSG	Message field. This is where status messages display.

HEX-DATA	Hexadecimal data. This is 1 to 4 columns of hexadecimal data.
CHARACTER	Character data. This is 1 to 16 characters of corresponding character data with a delimiting asterisk at either end.

The following primary commands are supported for the ZAP function. For information on ISPMODE commands available in StarTool FDM, see "Common Commands" in Appendix 2, "ISPF Interface Commands".

APP[LY]	Applies the specified line command to all table entries and executes each entry.
	Syntax `APPLY linecmd`
BA[SE]	Resets the base address for this CSECT (like an AMASPZAP BASE statement).
	Syntax `BASE hexaddress`
	(1 to 6 hexadecimal digits)
BU[ILD]	Outputs AMASPZAP format output without changing the data set. AMASPZAP backout controls are also formatted as comments.
EDITT[BL] (or ET[BL])	Enters an edit session on ZAP table data.
EXPR[ESS]	Executes all entered line commands without pauses between individual commands.
F	Finds a string and positions the display start location.

	Syntax
	`F hexstring|'anystring' [ASIS|CHAR]`
	`   [FIRST|LAST|PREV]    [PREFIX|SUFFIX|WORD]`
	`   [HALFWORD|FULLWORD]`

IDR[DATA]	Supplies the IDRDATA for the current ZAP.
	Syntax `IDRDATA idrname`
L[OCATE]	Positions to the address specified.
	Syntax `LOCATE address [BASE|ADDRESS|CSECT]`
NOZ[AP]	Exits the ZAP function without changing any additional data on disk.
OFF[SET]	Resets the beginning offset for CSECT displays. Lower offsets are not displayed.
	Syntax `OFFSET hexaddress`
	(1 to 6 hexadecimal digits)
O[PTIONS]	Provides primary command selection for the ZAP function and operand syntax assistance.
OUT[PUT]	Outputs the ZAP table to print or a data set.
	Syntax `OUTPUT [=c	F(ddname)]`
REM[OVE]	Trims the ZAP table based on a string match.

	Syntax
	`REMOVE anystring [PREFIX|SUFFIX|WORD] [NOT]`
	`        [HALFWORD|FULLWORD]`

RF[IND]	Finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.
TAG	Applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
	Syntax `TAG linecmd`
UT	Selects the extended user command panel. Commands are maintained in these panels by the StarTool FDM developers as well as your installation. The ZAP UT panel (actually the same panel as the log) allows dynamic primary commands with which you specify command names and their corresponding actions.
X	Clears the ZAP table relative to the cursor position.
	Syntax `X [ABOVE|BELOW|ALL]`
XA[LL]	Clears the ZAP table; this is equivalent to X ALL.
ZAP	Updates data lines in *CHA (changed) status on disk permanently.

The following line commands are supported in the ZAP function:

=	Repeats the previous line command. `*cm ERR- t="Line command,=">`
DIS	Disassembles a few lines starting at the current offset. Follow DI or DIS by an even number in the range of 2 through 14 to designate at what decimal offset the disassembly is to begin. For example, DI10.
K	Kills and clears all following line commands.
LOG	Copies the line into the log.
M	Provides line command selection and entry assistance.
O	Provides line command selection and operand syntax assistance.
TAG	Marks this table entry with *TAG* in the DATA/MSG field.
UNDO	Changes this data back to its original value. This does not restore data from before a ZAP primary command.
X	Drops the table line.

Block line commands are doubled letters that delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the ZAP function:

==	= command. Repeats the previous line command for the range of lines.
LL	LOG command. Copies the range of lines into the log.
OO	O command. Provides line command assistance for each line.
XX	X command. Drops the range of table lines.

```
 FUNCTIONS  CONTROL  LINE CMDS  DEFAULTS  FEATURES
-------------------- ZAP Display for DSAT-DSAT         ROW 78 TO 100 OF 458
COMMAND ===> build                                    SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=DSA* -----------------------------
          BASE = 000000   ADDRESS OFFSET = 000000
CMD MODULE BASE   MSG  ---------- HEX DATA --------------   -- CHARACTER --
    ADDR   OFFSET      0...2... 4...6... 8...A... C...E...   0...4...8...C...
    0004D0 0004D0 *CHA B7CA4100 000A4780 C23249F0 C60C4770  *........B..0F...*
    0004E0 0004E0 *CHA B512D507 D472C57C 4770B7CA D507D484  *..N.M.E@....N.Md*
    0004F0 0004F0      C5804770 B7CA5810 D214D21F 1000C9D8  *E.......K.K...IQ*
    000500 000500      5820D210 D22B2000 D2354520 C48047F0  *..K.K...K..D..0*
    000510 000510      BC7C49F0 C60A4100 00144770 C3B6D401  *.@.0F.......C.M.*
    000520 000520      904A904A 4100001E 4770C3B6 9610D22C  *.`.`......C.o.K.*
    000530 000530      D205D579 D5735810 D5B05010 D5845010  *K.N`N...N.&.Nd&.*
    000540 000540      D1B44121 000A5021 0000D2FD 2000D472  *J.....&...K...M.*
    000550 000550      4140D235 41440001 95404000 4770B554  *. K.....n  .....*
    000560 000560      5040D588 47F0B5A0 41440001 95404000  *& Nh.0......n  .*
```

```
 FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------- ISPMODE Session# 2 Log  ROW 1,008 TO 1,015 OF 1,015
COMMAND ===>                                          SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=DSA* -----------------------------
 NAME DSAT     DSAT                       NAME DSAT     DSAT
 VER  0004D4 000A47F0                     VER  0004D4 000A4780
 REP  0004D4 000A4780                     REP  0004D4 000A47F0
 VER  0004D8 C3B649F0                     VER  0004D8 C23249F0
 REP  0004D8 C23249F0                     REP  0004D8 C3B649F0
 VER  0004E4 D472C578                     VER  0004E4 D472C57C
 REP  0004E4 D472C57C                     REP  0004E4 D472C578
 IDRDATA   BUILDTST                       IDRDATA   BUILDTST
 ***************************** BOTTOM OF DATA *****************************
```

Appendix A

Formatting Member Names

StarTool FDM allows many different member name forms. Type member names in character or hexadecimal mode for any subcommand that processes member names.

Character member names can be from 1 to 8 bytes long with no imbedded blanks, commas, parentheses, colons, slashes, asterisks, question marks, or percent symbols.

Hexadecimal member names can contain from 1 to 16 hexadecimal digits delimited by x' and '. X'd7c4e2c5'. PDSE are entirely equivalent; also, x'333' and x'0333' are equivalent.

Subcommands ABE, ACFCOMP, BROWSE, COMBINE, COMPDIR, COPY, DCF, DELETE, DELINK, DIRENTRY, DISASM, DUP, EDIT, EXCLUDE, FIND, FSE, HISTORY, IF, LIST, LLA, MAP, MEMBERS, MEMLIST, OUTCOPY, PGMDOC, PRINT, READOBJ, REPLACE, REPRO, REVIEW, SMPGEN, SPFEDIT, SUBLIST, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT, and XREF allow additional member name specifications.

A member name range, member name pattern, or member name combination form what is termed a **member group**. A simple member name, a pattern, or a combination name can contain **?** or **%** characters as placeholders. You can also use a member list (that is, a parenthesized list containing member and member group specifications separated by commas).

When a member group is being processed, the subcommand processing routine is re-executed for each member in the group. Specifically, note the following:

Default member name	If no member name is specified, the last member name or member group entered for any of the above subcommands is assumed.
Asterisk member name	If you enter * in the member name position, the last member name or member group entered for any of the above subcommands is assumed.
Equal member name	If = is entered in the member name position, the list of members in the current MEMLIST table is assumed. If no MEMLIST table is active, the current member group is assumed (as for *****).

Member name range	Any of these subcommands can select all members in a range by specifying a beginning partial member name, a colon, and a ending partial member name (**either** or **both** of the partial names may be null).
	Logic from the DISPLAY subcommand selects members. The subcommand processes these selected members in alphabetic order. The following specifications select the same member names:
	<pre>subcommand : and DISPLAY
	subcommand part1: and DISPLAY part1
	subcommand :part2 and DISPLAY x'00' part2
	subcommand prt1:prt2 and DISPLAY prt1 prt2</pre>

Member name pattern	Any of these subcommands can select all members whose names contain pattern characters by specifying a member name segment, a slash, and another member name segment (either, **but not** both, of the member name segments can be null).
	Logic from the PATTERN subcommand selects members. The subcommand processes these selected members in alphabetic order. The following specifications select the same member names:

	<pre>subcommand seg1/ and PATTERN seg1
	subcommand /seg2 and PATTERN seg2
	subcommand seg1/seg2 and PATTERN seg1 seg2</pre>

Member name combination	Any of these subcommands can select all members whose names contain a range of characters and a pattern of characters by specifying a member name segment, an asterisk and another member name segment (either, **but not** both, of the member name segments may be null).
	Logic from DISPLAY and PATTERN subcommand selects members. The subcommand processes these selected members in alphabetic order. The following specifications select the same member names:

	<pre>subcommand seg1* and DISPLAY seg1 seg1
	subcommand *seg2 and PATTERN seg2
	subcommand seg1*seg2 and DISPLAY seg1 seg1 --with-- PATTERN
	        seg2</pre>

Member name placeholder	A simple member name, a member name pattern or a member name combination can contain a **?** or **%** character as a wild card character in any position.
	Logic from DISPLAY and PATTERN subcommand selects members. The subcommand processes these selected members in alphabetic order. The following specifications select the same member names:

	<pre>subcommand ab? and DISPLAY x'c1c20040' x'c1c2ff40'
	subcommand ab?* and DISPLAY x'c1c200' x'c1c2ff'
	subcommand *a?c and PATTERN a?c
	subcommand /a%c and PATTERN a%c
	subcommand a%c/xx and PATTERN a%c xx</pre>

Member group examples	The following examples each define and display a member group:
	
	``
	SUBCOMMAND MEMBERS IN THE GROUP
	---------- --------------------
	MEMBERS aa AA
	MEMBERS (aa,bb) AA and BB
	MEMBERS (aa,bb/) AA and members whose names contain BB
	MEMBERS * current member group
	MEMBERS current member group
	MEMBERS = members in the current MEMLIST
	
	MEMBERS : all members -- X'00' through X'FF'
	MEMBERS aa: members from AA... through X'FF'
	MEMBERS :bb members from X'00' through BB...
	MEMBERS aa:bb members between AA... and BB...
	MEMBERS (abc,d:) member ABC and those from D... through
	    X'FF'
	
	MEMBERS aa/ members whose names contain AA anywhere
	MEMBERS /bb members whose names contain BB anywhere
	MEMBERS aa/bb members whose names contain AA and BB
	MEMBERS (aa/,bb/) members whose names contain AA or BB
	
	MEMBERS aa* members with names AA...
	MEMBERS *bb members whose names contain BB anywhere
	MEMBERS aa*bb members with names AA... and BB elsewhere
	MEMBERS (aa*,bb/) members with names AA... or with BB
	    anywhere
	
	MEMBERS aa? members with names AA. (three characters
	    long)
	MEMBERS b?b members with names B.B (three characters
	    long)
	MEMBERS %a*b?d members with names .A and B.D after that
	MEMBERS ?a/b%d members with names .A and B.D in the name
	MEMBERS (aa?,?b/) members with names AA. or with .B anywhere
	``

Member name displays	Subcommands MEMBERS, PATTERN, and DISPLAY allow the display of member groups using the above member group syntax; these subcommands differ as follows: ■ DISPLAY – cannot modify the current member group. If no member specification is entered, the entire member directory displays. Member list notation cannot be used ■ PATTERN – cannot modify the current member group. If no member specification is entered, the member group last entered on a PATTERN subcommand displays. Member list notation cannot be used ■ MEMBERS – can modify the current member group. If no member specification is entered, the current member group displays. MEMBERS is one of the subcommands that define a member group. Member list notation can be used
Member name matching	StarTool FDM supports asterisks (*) in member name matching in the same manner as ISPF performs this function. StarTool FDM also supports underscores (_) in member name matching. ■ MEMBER a*test matches ATEST and A234TEST, but not ATESTX or ATESXT. ■ MEMBER a_test matches ATEST, A234TEST, and ATESTX, but not TESTA or ATESXT.

Appendix B

Update Protection

Several StarTool FDM subcommands can modify data sets when they are allocated as shared.

If the data set is allocated as OLD, the STOW DCB is left open after the first data set update. For OLD allocations, exclusive use of the data set is assured and the following RESERVE logic does not apply.

If the data set is allocated as SHR (SHR was specifically requested or SHR was defaulted to), StarTool FDM subcommands that update the data set such as ALIAS, (with attributes to be changed), COMBINE, COPY, DUP, DELETE, FIXPDS, RENAME, REPRO, RESTORE, SEPARATE, or VERIFY (for update member PDS tests) could cause data set integrity problems if the data set is also being updated by another user.

To circumvent this problem, StarTool FDM uses the following ISPF RESERVE logic to maintain data set integrity during data set updates.

1   Perform a `RESERVE (SPFEDIT,DSNAME,E,44,SYSTEMS),UCB=ADDRESS`.

2   If the data set is a load library, then a linkage editor RESERVE or ENQUE is performed as appropriate:

■   On a shared DASD volume,
`RESERVE (SYSIEWLP,DSNAME,E,44,SYSTEMS),UCB=ADDRESS`

■   On a non-shared DASD volume,
`ENQ (SYSIEWLP,DSNAME,E,44,SYSTEM)`

3   The data set STOW DCB opens.

4   All data set updates are performed.

■   Wherever possible, the STOW DCB is left open while an entire member group is being processed

■   For executions of StarTool FDM in batch mode, the volume reserve is maintained and the STOW DCB is left open following the first update to the data set

5   The data set STOW DCB closes.

6   DEQs remove the RESERVEs and ENQs*cm.

Appendix C
Dialog Errors

StarTool FDM displays a special panel when it encounters an ISPF dialog error. This panel formats an error message, highlights important registers, and presents recovery options.

```
------------------------------ Dialog Error ------------------------------
OPTION ===>

ISPP100  PANEL 'PDSXXZXX' ERROR
PANEL NOT FOUND.

Registers:
    R0 : 0000000C     R1 : 000E8748    *R2 : 9005ADAA     R3 : 5005459E
   *R4 : 0005AD82     R5 : 00053E10    *R6 : 00163738     R7 : 00088000
   *R8 : 00052E10     R9 : 00010000     R10: 00011000     R11: 00012000
    R12: 0003DF88     R13: 00164738     R14: 4005ADF8    *R15: 0000000C

Service name  ==> DISPLAY   (this may not be padded with blanks)
    First operand ==> PDSXXZXX

Options: ENTER - Continue if possible
         END   - Terminate (for some errors, several END's may be required)
         AB    - ABEND and attempt to clean up
         LO    - Display log table
         HELP  - Provide more information
```

```
TUTORIAL   ------------------------ Dialog Error Help -----------------------
OPTION  ===>

                    ------------------------------------
                    |           Dialog Error           |
                    ------------------------------------

StarTool has encountered an unexpected error in its dialog processing.

This type of error causes the Dialog Error panel to be displayed to document
the error and possible recovery options. When you get this type of error,
you should capture the error panel image with a PRINT command.

Normally, this type of error indicates a StarTool problem; you should report
this error to SERENA StarTool support.  Call SERENA at (650) 696-1800.

The Dialog Error panel is formatted as follows:
    Error message: Message identifier followed by short and long descriptions.

Error registers: Important registers are hilited as follows:
      *R2  - Return address for caller of lowest level routine
      *R4  - Base for lowest level routine
      *R6  - Base for dialog work area
      *R8  - Main base for dialog
      *R15 - Return code of the failing service

Service name: This is the name of the ISPF service that failed.

First Parameter: This parameter was passed on the ISPF service call.

Options: This identifies different options for dealing with the error.
      ENTER - Continue if possible; in most cases, the error can be ignored.
      END   - Terminate StarTool; this is the equivalent of a QUIT subcommand.
      AB    - ABEND; current processing will be suspending to go to line mode.
      LO    - Display log table; this option ignores any current processing.
```

Appendix D
Attention Processing

When StarTool FDM is executed as an ISPF dialog, attentions are not required since an output loop is detected by the checkpoint processing of ISPMODE. If StarTool FDM or a supporting TSO command is in a CPU loop (with no output), you need to interrupt the process with an attention key or it continues indefinitely.

In general, an attention (the PA1 key) terminates the current StarTool FDM subcommand. Two attentions in a row terminate the **STARTOOL** command. StarTool FDM performs attention processing as follows:

1 A single (double for TSOEDIT) attention terminates any member group in progress (except for the FSE subcommand).

2 An attention at a prompt for a data set terminates StarTool FDM.

3 An attention at the normal PDS300A ENTER OPTION subcommand prompt provides another ENTER OPTION prompt; another attention at this point terminates StarTool FDM.

4 For subcommands BROWSE, EDIT, ISPF, ISPMODE, or MEMLIST, no ENTER OPTION prompt is provided by the StarTool FDM attention handling:

- A single attention terminates any member group in progress

- An attention during a edit or browse line command from MEMLIST terminates any remaining line commands

- Any additional attentions are ignored by StarTool FDM

- For TSO commands invoked by ISPF with no attention handling, the first attention is noted by StarTool FDM and terminates any member group in progress; subsequent attentions apply only to the TSO command

5 For the COMPRESS or COPY subcommands, attentions are ignored until the attached program terminates to protect the integrity of the target data set.

> **NOTE** If you use IKJEFTSR authorize IEBCOPY for the compress or copy, attention interrupts cannot be deferred; thus, do not use the attention key during a compress or copy (a successful attention can destroy the data set).

6 For other StarTool FDM subcommands, a single (double for TSOEDIT) attention results in a PDS300A ENTER OPTION prompt. The processing action taken at this point depends on what is entered next:

- Another attention terminates the StarTool FDM command

- A null line causes the interrupted subcommand to continue at the point of interruption. Any messages awaiting output at the time of the interrupt have already been discarded.

- If you enter a subcommand and the currently executing subcommand is external to StarTool FDM (ABE, ACFCOMP, COMPARE, DCF, EXEC, FSE, HELP, PRINT, REVIEW,

SUBMIT, TSO xxx, TSOEDIT, TSOLIST or VPRINT), the subcommand processor forcibly detaches and the ABEND message is suppressed

■ Give an entered subcommand control without delay

7 If an internal StarTool FDM subcommand is in a loop that does not involve input or output to the terminal or input from the current data set, a single attention does not terminate the loop:

■ The first attention provides the normal PDS300A ENTER OPTION prompt.

■ If you enter a subcommand at this time, StarTool FDM continues looping.

■ Another attention produces the message "PDS470W The program is probably in a loop" and StarTool FDM performs the subcommand entered at the ENTER OPTION prompt.

ABEND Processing

IF an ABEND in a StarTool FDM module, the **StarTool ABEND** panel displays:

```
------------------------------ StarTool ABEND ------------------------------
OPTION ===>

PDS999E ABEND S0C1 U0000 AT +005F3E IN PROGRAM PDSPARSE
              Offset of error: 00036CA6
   Address of next instruction: 00046CA6

Registers:
    R0 : 00000000     R1 : 00000003     R2 : 00000000     R3 : 000007AB
    R4 : 00178000     R5 : 00000000     R6 : 00022230    *R7 : 00088000
   *R8 : 00021230    *R9 : 00010000     R10: 00011000     R11: 0001200
    R12: 0003DF88     R13: 0008933C     R14: 40021C68     R15: 00046C9C

Press HELP for more information
```

Press the HELP key (PF1) to bring up the **ABEND Help** panel.

```
TUTORIAL   ----------------------- StarTool ABEND Help ----------------------
OPTION  ===>
                       ------------------------------------
                       |          StarTool ABEND          |
                       ------------------------------------
StarTool has ABENDED as indicated on the previous panel.

When an ABEND is encountered, StarTool displays the ABEND panel to document
the error and display error registers.  When you get this type of error, you
should capture the error panel image with a PRINT command.

Normally, this type of error indicates a StarTool problem; you should report
this error to SERENA StarTool support.  Call SERENA at (650) 696-1800.


The StarTool ABEND panel is formatted as follows:
    Error message: PDS999E ABEND Sxxx Unnnn AT +hxval IN PROGRAM progm
      Sxxx  - the system ABEND code
      Unnnn - the user ABEND code
      hxval - if signed, an offset from the routine entry point; otherwise,
              the address of the abending instruction.
      progm - the name of the abending program if available.

   Offset to error: This is the displacement from the start of the program to
      the instruction that failed.  This offset could be used to disassemble
      the instructions at an error as in the following example:
          DISASM STARTOOL OFFSET(1214C)

   Address of next instruction: This is the storage address of the
      instruction just after the instruction that failed.

   Error registers: Important registers are hilited as follows:
      *R7 - Base for the main work area
      *R8 - Base for the current subroutine
      *R9 - Main base for StarTool
```

StarTool FDM always attempts to recover from ABENDs with its ESTAE and ESTAI recovery routines unless CONTROL NORECOVER is in effect. StarTool FDM performs ABEND processing as follows:

1 The StarTool FDM program uses different recovery methods depending on the subcommand being executed. The method used performs any cleanup and reinitialization functions required to restart StarTool FDM.

■ For internal StarTool FDM subcommands, recovery is performed by writing an error message and terminating the subcommand

■ For the external subcommands BROWSE, EDIT, or ISPF, no ESTAE recovery is attempted; normal ISPF processing is permitted

■ For the other external StarTool FDM subcommands (ABE, COMPARE, COMPRESS, COPY, DCF, EXEC, FSE, HELP, PRINT, REVIEW, SUBMIT, TSO xxx, TSOEDIT, TSOLIST, and VPRINT), the subcommand processor is forcibly detached

2 If CONTROL NORECOVER is in effect, StarTool FDM terminates unless the subcommand being used specifically tests for ABEND conditions (such as IF with LOADERR or VERIFY with LOAD).

3 If CONTROL RECOVER is in effect, any ABEND dump probably is not very useful. If you want to use TSO TEST after an ABEND, set CONTROL NORECOVER first.

4 Certain external programs reached with the TSO subcommand (such as CALL) terminates StarTool FDM with a READY message if the program they invoke ABENDS. If you enter a null line at this point, StarTool FDM regains control.

5 The PDS999E message identifies the name of the StarTool FDM assembly listing involved in the error as follows:

PDSMAIN	ABEND in the StarTool FDM mainline
PDS#SECI	ABEND in the security interface
PDSALIAS	ABEND in the subroutine assembly
PDSCBSX	ABEND in the Copybook setup routine
PDSCPARS	ABEND in the Copybook parse routine
PDSDECOD	ABEND in the DISASM routine
PDSDELNK	ABEND in the DELINK routine
PDSFCALC	ABEND in the CALC routine
PDSFILE	ABEND in the StarTool FDM execution interface
PDSFPARS	ABEND in the StarTool FDM emulation parser
PDSIDCAM	ABEND in the IDCAMS interface
PDSIDSPY	ABEND in the ISPMODE dialog
PDSIPARS	ABEND in the StarTool FDM parser
PDSPARSE	ABEND in the PARSE interface
PDSSPACE	ABEND in the service routines
PDSVTOCR	ABEND in the VTOC read routine
VTSOCMD	ABEND in the TSO command check

Glossary

A

Alias member A member whose directory entry has an indicator 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 are available. In this mode of operation, StarTool FDM operates under control of the batch Terminal Monitor Program (the TMP, or IKJEFT01).

BLDL A BPAM function that checks for the existence of members.

BPAM An acronym for Basic Partitioned Access Method; BPAM is actually very similar in use to BSAM which is for sequential data. StarTool FDM often uses BPAM for member check operations and always uses BPAM for directory updates and reading PDSE data sets.

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

C

CCHHR An acronym for Cylinder, Cylinder, Head, Head, Record which is a 10-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.

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 any 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 An acronym for 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,...) as well as 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 commonly used to contain 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 contained in a load library. Also known as a load module or load member.

O

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

P

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

PDSE or Partitioned Data Set Extended   An SMS managed data set 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 which looks like a PDS.

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

Q

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

S

Source library   A PDS or PDSE data set with fixed or variable format data; source libraries are used to contain non-executable data.

StarTool FDM   A multipurpose ISPF dialog and TSO command processor that manipulates data sets and members individually or in groups.

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.

T

TTR	An acronym for 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

Symbols

= line commands 51
== block line commands 51

A

ABE subcommand 51
ACFCOMP subcommand 52
add secondary space 188
alias members 58
ALIAS subcommand 53
asterisks
    in member name matching 423
attention processing 429
ATTRIB subcommand 59

B

binder names 63
BLK3380 subcommand 63
BLK3390 subcommand 65
BLK9345 subcommand 68
Block line command
    AA 128, 241, 248, 258, 295, 413
    BB 128, 241, 248, 258, 295, 413
    CC 84, 128, 241, 248, 258, 320, 390, 413
    DD 84, 295, 320
    DDEL 248, 258, 295
    EE 128, 241, 248, 258, 295, 413
    GG 128, 241, 248, 258, 390, 413
    HH 295
    LL 77, 121, 128, 241, 248, 258, 266, 295,
        390, 413, 419
    MM 84, 128, 241, 248, 258, 295, 320, 390,
        413
    OO 77, 121, 128, 241, 248, 258, 266, 295,
        300, 390, 413, 419
    PP 295
    RR 84, 320, 413
    SS 77, 121, 128, 241, 249, 258, 266, 295,
        390, 413
    SSEE 128, 241, 249, 258, 413
    TT 295
    UU 128, 242, 249, 258, 413
    VV 295, 413
    XX 77, 84, 121, 128, 242, 249, 258, 266,
        295, 300, 390, 414, 419
    ZZ 128, 242, 258, 414

block line commands 77
    defined 51
BROWSE subcommand 70

C

CALC function 73
CAX function 74
    line commands 76
CAX table
    catalog status 75
    catalog type 75
CHANGE subcommand 78
checkpoint 271
CLIST conversion 110, 145
CMDTBL function 80
    line commands 83
    primary commands 82
    table entries 82
combination member name 30
COMBINE subcommand 85
Command
    ? 37, 40
    ADD 40, 264, 409
    ALIASCHK 37, 40, 291
    ALL 37, 40, 291
    ALLOCERR 37, 40
    ALTCMD 37, 40
    ALTERNAT 37, 40, 125, 239, 244, 254, 264,
        272, 291, 330, 388, 409
    APPLY 37, 40, 76, 82, 119, 125, 239, 244,
        254, 264, 292, 299, 388, 409, 418
    BASE 41, 418
    BATCHJCL 41, 125, 239, 272, 292
    BLKSIZE 37, 41
    BOUNDS 41, 316
    BUILD 41, 418
    CAN 37, 41, 272
    CANCEL 41, 317
    CAX 76
    CB 41, 317
    CBMAP 41, 317
    CHANGE 41, 317
    CISIZE 37, 41
    CLOSE 41, 317
    COLS 41, 42, 272, 310
    COMMENT 37, 42
    COMPRESS 42, 125, 239, 244, 254, 409
    CONTINUE 42, 272
    COPYBOOK 42, 317

M

N